CHAPTER 2

RELATED RESEARCH WORK

This chapter presents the main area of research related to the target scenario presented in chapter 1. This scenario exhibits two main challenges: routing to sink(s) with failure management; mobility; and energy-efficient, low-overhead, possibly non-uniform clustering. The chapter will discuss each of them individually, explain why it is extremely difficult to solve them and outline related efforts for meeting them.

Additionally a survey on machine learning (ML) and computational intelligence (CI) applications in wireless sensor networks is presented. Our first intuition is that this class of algorithms is highly suitable to meet all of the challenges of the presented application scenario. However, it needs to be understood how each of these algorithms performs in the context of wireless sensor networks in order to identify the best suited technique for this thesis.

Routing In WSNs

There is a large research body on routing protocols for WSNs, based on various assumptions, cost metrics and network scenarios. Simple techniques like Mint Route or Directed Diffusion are usually preferred. Recent research is focusing on energy-aware routing protocols. One of the widely used routing protocol in this class being Low-Energy Adaptive Clustering Hierarchy (LEACH) and its variants. Work on this protocol shows increase in network lifetime by 20-30% over a standard S-MAC Protocol.
Most efforts concentrate specifically on particular challenges or problems, but none of them gives a general, flexible and robust solution to all of them simultaneously. The goal of this dissertation is to design and implement a general solution to all these problem applying machine learning techniques.

**Clustering in WSNs**

Similarly there exists a wide variety of clustering protocols for WSNs, and efforts are directed on finding the optimal clustering scenario. Clustering together with data aggregation has been shown to inherently decrease the communication overhead in WSNs, to save energy and to improve delivery rate (for instance the LEACH protocol). However, there are same major challenges not yet efficiently met. The first challenge of clustering protocol is the process of clustering selection. It can be demonstrated that clustering protocol based on Machine Learning technique is more efficient than the random selection process of LEACH. The Machine Learning based clustering protocol would reduce the communication overhead of cluster-based selection, be robust against node failures and use energy in an efficient way. Besides it would support non-uniform data requirements.

**2.1 Energy efficient multicast routing in WSN**

While a large body of different routing protocols has emerged in the last years, there is still no general and well-performing routing protocol for WSNs. Real deployments often decide for a simple, already implemented routing protocol based on hops like MintRoute [20] for TinyOS. However, they often also change the protocol according to their needs [21, 22, 23], for example by using a different neighborhood management protocol or a suitable customized cost metric. Thus, the resulting protocols are highly specialized and optimized solutions for the targeted network rather than a standard protocol for a broad variety of scenarios.
In this chapter is given an overview of state-of-the-art routing protocols. First, we summarize traditional point-to-point routing algorithms before proceeding with multicast approaches for WSNs. An in-depth survey in terms of sink mobility, failure management and various routing cost metrics is also given.

2.1.1 Point to point routing in WSNs

There are many different techniques and routing protocols, which are adopted for WSNs and several surveys, have tried to classify and summarize them [24, 25, 26]. Several different routing protocols have emerged from routing protocols for Mobile Ad Hoc Networks (MANETs). They build a full routing path table at all nodes and for each possible destination each node maintains a full route. The main drawback of such an approach is that routing information needs to be propagated throughout the network (from the source to the destination and back). Secondly, in case of changes in topology or failures to re-build the routes, a complicated route repair procedure needs to be started. Some protocols take an abstraction step of dividing the route into segments, where only segments are required to be repaired [27]. MANET-based protocols have been implemented in WSNs with some changes (in this case, multi-path routing), like AOMDV [28] based on AODV [29]. However, the main disadvantages still exist.

One of the popular routing techniques designed especially for energy restricted unreliable wireless sensor networks is content based networking [30]. In this routing framework, the data is sent from the source to the destination, which is like demand, and supply, based on interests expressed by the destinations to receive a particular pattern of data. Such an approach is relevant for sensor networks as it is data driven as opposed to address driven. This has been demonstrated in [31] where the authors use a distance
vector protocol to construct a tree from the source node to an interested sink.

Another application of content based networking for sensor networks is Directed Diffusion [32, 33] where routes from the source to the destinations are established on demand based on interests that are flooded through the network. This flooding establishes gradients for data to follow to the sinks from multiple sources. As the source sends low rate data samples, the routes where data first arrives are reinforced by the sinks.

Directed Diffusion motivated many other routing protocols. Rumor routing [34] and its successor, Zonal rumor routing [35] limit the initial interest propagation phase by routing the interests only to the specified zones in the network. For this, each node needs to be aware which node is producing what kind of data. When a node produces data, it generates a long lived agent, which traverses the network and goes to every other node and informs them about the available information.

GRE-DD [36] and LMMER [37] belong to the primary category and are each extensions of Directed Diffusion [11]. They take into account the remaining battery level of neighbors once choosing the gradient to the sink. However, they are doing not dynamically modifying the gradient, though a node exhausts its energy. Instead, they must wait till the next sink flooding to update the battery level and therefore the route. An identical approach is delineated in [38], where every node is aware of the “heights” of its neighbors (number of hops to the sink). If the battery level of some node deteriorates, it will increase its height and broadcasts this new information to its neighbors.

MintRoute [20] from TinyOS\(^1\) is a similar hop based routing approach, which additionally incorporates a neighborhood management protocol. It selects the next hops based on link quality and hops to the sink.
Positional knowledge of the nodes is the basis of location-based (or geographic) network routing. The nodes of the network are able to obtain either their exact coordinates by a GPS enabled receiver or their relative locations by strengths of the incoming signal from their neighboring nodes. For example, GEAR [39] is an improvement over Directed Diffusion, where interests are routed to their destinations via a location-based heuristic. Thus, flooding of the interests is restricted which saves a lot of energy.

A traditional geographic routing protocol is GPSR [40], which selects next hops depending on their advance to the sink. In case the routing is stuck (a node is reached with no progress to the sink), a unique face routing process is started to route the packet around the void region. The main drawback of geographic routing protocols is the length of the selected routes, especially in case of void regions. An effort to overcome this problem is presented in [41] which describes a landmark assisted geographic routing protocol. Here, in a pre-processing phase the nodes exchange information about their location and the full global topology is reconstructed at each of the nodes. However, topology information is abstracted and the network is divided into tiles. Thus, without full-network broadcasts of the events, node failures and low mobility can be handled. Special landmarks are used for routing the packets through the tiles. Unfortunately the work is not evaluated in terms of overhead or spent energy and no comparison to existing works is given.

![Figure 2.1. A typical architecture with two destinations, the primary paths to them from source S and its initial routing table.](image)
Another area of concern with traditional geographic routing schemes is their preference for long unreliable hops. If no separate link protocol is used, geographic routing selects next hops based on their progress to the sink only, thus, mostly long loss connections are activated. An extensive study of this problem and a comparison of various other location based metrics used in simulation and real hardware is presented in [42]. Blacklisting highly unreliable neighbors are compared with traditional greedy strategies, selecting only the most reliable neighbors and using the product of geographic progress and reception rate to identify the next hop. The study shows the last product-based metric results in highest end-to-end delivery rate.

2.1.2 Multicast routing in WSNs

Let us consider the sample topology from Figure 2.1. It shows a small network with two sinks P and Q and a single source S. After the proposed sink announcement, all of the nodes in the network have some initial routing information, e.g. hops to the individual sinks. For example, node S (the source) has three neighbors and routes to both sinks through each of them. According to its information, the next best hops for sink P include neighbor A and neighbor C for sink Q. From the point of perspective of the source it looks like the route costs (3 + 3) hops or 5, if the first hop is shared through a broadcast message (the dotted route in the figure).

However, by considering the network graph globally, it is immediately observed that the route through nodes B, F, H and then to the sinks (the middle route in the figure) costs only 4 hops - even in this small network there is possible saving of 20% compared to the best locally available route. Additionally, the remaining energies on the nodes can be considered. Finding the globally optimal route is what is called the multicast challenge.
There are various numerous different approaches available on solving the multicast challenge. Many traditional multicast routing protocols come again from the MANET environment, for example MAODV [144], LAM [94], AMRIS [203], ADMR [93], and RBM [43]. They build a multicast tree in the network on demand, through the exchange of control packets. However, this approach requires large overhead for building and maintaining the tree, especially in conditions of failures and mobility. There are some recent works using swarm intelligence [51, 167], but again the overhead from sending ants is not bearable for WSNs. Other researchers have also reported about substantial challenges and problems when implementing MANET multicast routing protocols for sensor networks, like the implementation of ADMR on MicaZ motes [37].

Mesh-based algorithms for MANETs maintain an overlay structure for forwarding data to each and every receiver. They have proven to be quite efficient in high mobility scenarios, but cause great overhead in communication for constructing and maintaining the mesh and thus cannot be successfully applied to WSNs. Such protocols are for example ODMPR [115], CAMP [68], PUMA [191], AMRoute [207], and PAST-DM [74].

From the WSNs community there are two main groups of research efforts in the area of multicast routing: geographic based and "fake multicast". GMR [56] and MSTEAM [57] are both geographic based multicast routing protocols. These approaches do not need any control packet exchange to build the multicast tree. In fact, they take next hops greedily to reach the sinks. However, having geographic information in large, randomly deployed sensor networks are far too unrealistic or quite costly, and thus suitable alternatives have to be found. Another major drawback of geographic protocols is the so called face routing, which is used for routing data around
face (void) regions. This takes a much longer route than required and does not learn from its previous experience, like previous routing around the same void area.

Another approach for WSNs for multicasting is what is usually called as "fake multicast": unicast protocols, which are slightly optimized for multicast routing. Such protocols only construct routes from an origin to every of the destinations not really finding globally optimal ones or considering sharing of paths - a simple example of which is Directed Diffusion [58], which easily supports multiple sinks. Another work [59] emphasizes on sharing of paths from many origins to many destinations by locally sharing subsequent hops at the same costs. However, the main assumption of [59] is that packets from different sources can be aggregated, which makes the work a tree based clustering approach rather than a traditional routing algorithm. Additionally, the definition of the routing cost function leads to routing oscillations in the beginning, causing a lot of additional overhead communication.

Other researchers [60] postulate the problem of multiple sinks routing in a different manner: it finds the optimal data rates of all sources of data in a WSN. In short, each source routes its data only to the next sink and all sinks cooperate to reconstruct the data field. Similarly, [61, 62, 63] present solutions to the optimal sink placement problem in a WSN. A study on the multicast capacity of certain networks is presented in [64].

Again, there are other mesh or overlay routing protocols, which successfully handle multiple mobile sinks. These are discussed in the following section.

2.1.3 Sink mobility in WSNs

Some routing protocols are based on the assumption that the mobility pattern of the sinks is known beforehand at the sensor nodes. One such protocol is the spatiotemporal mobicast routing algorithm in [65]. This protocol is exclusively an overlay routing
protocol, which decides when to further pass the information via a geographic routing protocol and to which neighbors. In this manner it ensures spatiotemporal receipt of needed info to the required region where it is needed. The work was further developed in [66], which is able to provide better handling of void areas. IDDA [67] is following a similar idea, where the mobile sink makes use of a directional antenna to wake up nodes in its next location. Nodes in the predetermined area use gradients as in Directed Diffusion to send data to the node next to the sink’s future location, thus preparing data for the sink and waiting for the sink there.

TTDD [68] is a layered routing protocol, developed especially for scenarios of high mobility. The researchers dwell on a cost effective receipt to many mobile destinations by building a routing overlay. The network is clustered into a number of cells and non-stationery destinations inundate their queries only in the nearby cell itself. Thus, the overlaid is always aware of the current position of the sinks and routes the data to them. This approach has proven to be quite effective in scenarios of high mobility. But, the nodes building the overlay (a cell structure) drain their power quickly and the overlay has to be rebuilt with high overhead communication. That is why the protocol is better suited for sensor networks for event detection with only irregular traffic rather than continuous monitoring. Other overlay based routing protocols are ODMPR [69], CAMP [70], PUMA [71], AMRoute [72] and PAST-DM [73].

SEAD [74] optimizes paths from one origin to many mobile sinks. Every sink selects an "access sensor node", to which data has to be routed from the source. A tree is built based on a geographic location heuristic between the source and all other access nodes. When the sink moves slightly away, a path between its current nearest neighbor and the access node is maintained, so that it is not necessary to rebuild the tree. If the sink moves too far away, a new access node is selected and the tree is built
again, but only with high overhead communication. The results are good compared to Directed Diffusion [75] or TTDD [76] with regard to energy consumed for data packets. But, no extensive appraisal of the control overhead under mobile sinks is presented, which is expected to be quite high. A further refinement of SEAD is DEED [77], which introduces delay constrains on the multicast routes.

Multiple mobile sinks are the target scenarios for DST [78]. A shared routing tree is constructed by the first (master) sink and shared by next slave sinks. Unlike SEAD [76], the whole tree is dynamically updated when sinks moves far away from their access sensor nodes. The approach shows slightly better results than SEAD in high mobility single-sink scenarios and the same performance as SEAD in multiple-sink settings. An analytical evaluation of virtual infrastructure routing protocols (TTDD [76], SEAD [79] and others) is presented in [80].

2.1.4 Failure recovery or routing in WSNs

One of the main challenges in routing is managing link and failures of node. Failures have been widely considered in routing for WSNs and different approaches have been taken. The most important design criterion is to be able to register a failure and to easily update the available next hops.

Failure recovery is closely related to and in fact an integral part of link quality management. Here, two different techniques exist: proactive beacons and passive refreshment of routes. The first technique is used by nearly all of the management protocols and by nearly all geographic routing protocols [81, 82, 83]. Here, the nodes exchange small non-data related packets (beacons) to refresh their information about their 1-hop neighbors. Usually, the RSSI (Received Signal Strength Indication) level of the radio signal and data reception rate is used to compute the quality of the link.
Failure recovery is automatically incorporated in these algorithms by assigning very low quality to failed links (non-responding nodes), and thus signaling the problem to higher layers. The main drawback of separate link management protocols is their unawareness of the requirements of the higher layers. For example, many link management protocols supply the higher layers with a list of "good" neighbors or a list of the n-best neighbors. In the first case, the routing protocol is not able to choose the best neighbor because of lack of knowledge, in the second case it might miss a good neighbor, which has a good quality, but resides on place n + 1 of the quality-sorted list.

The second recovery technique, passive refreshment of routes, is applied often by hop-based routing protocols like Directed Diffusion [75], which do not make use of any separate link management. Here, the sinks refresh the routing information at regular intervals by a full network broadcast of a simple control packet, called the sink announcement or data interest. This is an efficient and general approach to notify nodes in the network about the sinks’ requirements. However, sending such an announcement too often, for example to keep routes up to date, is not considered to be efficient and dramatically increases the data traffic in the network. A similar technique is also adopted by all MANET-like routing protocols, where control packets are exchanged at regular intervals to refresh routes.

2.1.5 Routing cost metrics for WSNs

Location-based (geographic) routing is probably one of the largest families of routing protocols. Here, progress to sink is used as routing cost metric and next hops are selected accordingly [81, 82, 83].

A metric coming from MANET routing protocols is end-to-end latency, as used for example in the original two phase pull version of Directed Diffusion [75]. Here, the sources start delivering data at low rates over many possible routes. Thus, the sink
observes from where the data arrives first and reinforces this route, which becomes the main route for delivery of data at higher rates.

In homogenous WSNs, the number of hops is inter-related to latency and has been used extensively as a metric for routing [84, 75]. Hops are a simplified version of latency and can often be used interchangeably with it. Both metrics have several advantages over location awareness: they are cheaper to acquire and they automatically build minimum hop/latency (shortest path) routes without void areas. On one side this leads to shorter, very energy-efficient routes. However, these routes quickly go into depletion and the network could become disconnected.

Therefore, other research efforts additionally take the residual energy of the node into account. Such approaches work in one of the two ways: considering strictly localized information where only the neighbors’ remaining energy is given [85, 86, 87, 88, 89], or full global information where all remaining power levels for all nodes are known at the base station [90]. Considering the remaining energy on the 1-hop neighbors has the advantage of being fully localized and thus very energy-efficient, but does not guarantee that the nodes on the remaining path to the destinations have high energy reserves. On the other hand, global information helps to identify truly optimal routes, but has a large overhead communication.

One of the widely used cost metrics for computing the quality of links and neighbors is the RSSI level of packets received, assuming that high RSSI values come from nearby, reliable neighbors. However, some researchers [91] use this metric also with the opposite assumption, i.e. low RSSI indicates a neighbor far away, and use it to select neighbors, which are possibly further away and thus closer to the destination. However, such a metric suffers from the same drawbacks as suffered by geographic routing - the connection link to the farthest neighbor is usually very prone to error,
which results in many retransmissions of the same packet or a high loss rate of packet. Another use of RSSI is the computation of the distance between the sender and the receiver and is often used by clustering approaches (Section 2.2).

A current effort to improve connectivity in wireless sensor networks has led to a new cost metric, the connectivity importance value [92]. A node is considered important if after failing it will disconnect part of the network. Thus, routes are taken which avoid important nodes to avoid disintegration of the network. Unfortunately, these values cannot be computed in a distributed manner, since information of the entire topology is needed on all of the nodes in the network. The values also need to be computed again after node failures, reflecting the new topology. This becomes a communication challenge especially towards the end of the network lifetime when nodes start to fail rapidly one after another.

2.1.6 Routing in WSNs: Summary

Large groups of researchers are working on routing protocols used in WSNs, based on various assumptions, cost metrics and network scenarios. Simple techniques like MintRoute [20] or Directed Diffusion [75] are usually preferred. But, they do not efficiently manage multiple sinks, failure of the nodes or mobility. Other efforts focus specifically on one of these challenges, but none of them gives a generalized, more flexible and robust solution to all of the problems simultaneously.

Thus, the goal of this dissertation is towards the design and implementation of a general solution to all of these problems. However, unlike the solutions presented here which need a substantial increase in processing, memory or overhead communication to handle each of the described challenges one by one, our solution needs to be universal and self-consistent.
2.2 Energy efficient clustering for WSNs

This method attempts to minimize global energy dissipation by forming clusters that is a grouping of nodes. As a starter members of a cluster elect a leader - cluster head (CH) and transmit the data to the CH, rather than the base station. The energy consumption is now reduced as the nodes transmit to a nearby CH. Clustering and knowledge aggregation have tried to be powerful techniques to minimize energy consumption in WSNs, whereas at a same time keeping some minimal characteristics of the sent knowledge.

Though it is a simple and straightforward approach, this approach hides away important and difficult to solve issues. In particular, the selection of cluster heads is critical: randomly selected heads do not cover the sensors well and cause a miss-match of intra-cluster communication overhead. Settled choice supported remaining energy, ID, or different metrics, needs either k-hops neighborhood information at all nodes or global information about the network to compute the optimal clustering. The announcement of cluster heads causes non data related overhead communication and failures of cluster heads which forces the entire cluster to fail.

This survey isn’t meant to be complete because it is not possible to do so in this report. There are however, six primary group of protocols: random, 1-hop grid,k-hop grid, location based and centralized bunch protocols. Their main properties and our classification of the protocols are summarized in Figure 2.2. Note that some of the protocols are marked as exceptions, because their properties differ in some way or the other from most of the other protocols of the same family. A discussion on these exceptions in detail together with the related works is next.

After presenting state of the art clustering protocols, a short summary of data aggregation techniques is given and concludes the survey with related efforts on
theoretical studies and optimal clustering techniques.

2.2.1 Random protocols

Several agglomeration rules are enhancements or alterations to LEACH [1]; cluster heads chosen on a priority probability. Designated cluster leaders broadcast a cluster head role communication to all concerned motes that successively determine and select the closest cluster head. This chance correlates to the required cluster leaders, the idea behind the original LEACH protocol. More metrics like residual mote energy [6, 7] may be applied to modify the clustering properties. [94, 95, 96, 97].

Two important advantages have been seen in random clustering algorithms. First as the CH selection is random control messages are not required for the nodes to converge to their assigned cluster head (CH) and secondly are simple and easy to implement. The sizes and shapes cannot be predicted due to the stochastic nature of the algorithm is its greatest drawback. Concretely data from half the network is aggregated and at times only a few are processed. The second bottleneck is an assumption that the communication is one-hop. So a significant control overhead is required in a multi-hop network. (See Figure 2.)

TEEN [98] and APTEEN [99] are built over LEACH and further reduce the number of packets transmitted by introducing thresholds on the gathered sensory data: if the threshold is not exceeded, the node does not inject a new data packet into the network. However, the clustering protocol remains the same as LEACH.

<table>
<thead>
<tr>
<th>CLUSTER HEADS</th>
<th>CLUSTER FORM</th>
<th># HOPS IN CLUSTER</th>
<th>TAXONOMY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random nodes</td>
<td>Random form</td>
<td>1 hop, var. trans.</td>
<td>RANDOM</td>
</tr>
<tr>
<td>LEACH [93],[94],[95],[96], TEEN [98], APTEEN [99],[97]</td>
<td>LEACH [93],[94],[95],[96], TEEN [98], APTEEN [99],[97] LNCA [100]</td>
<td>Power LEACH [93],[94],[95],[96], TEEN [98], APTEEN [99],[97]</td>
<td>PROTOCOLS</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Min/max distance from other CHs</strong></td>
<td><strong>Quasi-circular</strong> [101], BP [102], UCCP [103], TRC [104], [105], EDC [106], FLOC [107], PC [108], [109],[76],[110], LNCA [100], HEED [111],[112]</td>
<td><strong>1 fixed trans. Power</strong> BP [102], UCCP [103], PC [108],[76], HEED [111]</td>
<td><strong>1-HOP GRID CLUSTERING</strong></td>
</tr>
<tr>
<td><strong>Location-based</strong> [76], GROUP [113]</td>
<td><strong>Exact (squares, hexagons, etc.)</strong> [76], GROUP [113]</td>
<td><strong>any possible</strong> [114],[105],[102],[115],[120],[116],[117],[121],[118], GROUP [113]</td>
<td><strong>LOCATION AND TREE BASED CLUSTERING</strong></td>
</tr>
</tbody>
</table>

**Power**

1 fixed trans.

**Parameter k**

[101], [104],[105], EDC [106], FLOC [107], [109], [76],[110], LNCA [100], HEED [111],[112]

**k-HOP CLUSTERING**

**LOCATION AND TREE BASED CLUSTERING**

**INFRASTRUCTURE SUPPORTED**
<table>
<thead>
<tr>
<th>Tree-based</th>
<th>Geographically limited</th>
<th>CLUSTERING</th>
</tr>
</thead>
<tbody>
<tr>
<td>[114], [115]</td>
<td>[114], [115], [116], [117], [118]</td>
<td>NON-UNIFORM CLUSTERING</td>
</tr>
<tr>
<td>Pre-deployed</td>
<td>Non-uniform</td>
<td></td>
</tr>
<tr>
<td>[116], [117], [118]</td>
<td>[105], [119], [120], [121], [117],</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[105], [119], [120], [121]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.2. Classification of latest clustering protocols and their main properties.

### 2.2.2 1-Hop grid clustering

LEACH a primary routing protocol uses 1-hop clusters for the entire network. However such scenarios are not possible most of the times. So two families of protocols have evolved: 1-hop and k-hop clustering rules. Typical of the 1-hop grid clustering protocols are HEED discussed in [111], BP [102], Passive Clustering (PC) [108] and others [105, 112]. The process involves a random assignment of cluster heads to few nodes of the network and then groups are evolved around them. It is
reasonable to assume densely populated networks for 1-hop communication. The overhead control for agreeing on cluster heads is significant. As the communication is 1-hop the size of the clusters depends on the communication radius of the nodes and is generally semi-circular. To meet the criterion of 1-hop grid communication optimal clustering is usually defined to minimize the quantity of clusters meeting this (1-hop) criterion. It is essential for these algorithms to keep the quantity of groups as small as possible. Some of the 1-hop grids clustering approaches are location-based [76]. Here the size of the cells is selected in such a manner that communication from neighboring cluster heads to cluster heads is guaranteed.

### 2.2.3 K-hop clustering

When data transmission amongst cluster members and the head require more than one node in between it is multi-hop communication. This type of communication eliminates 1-hop virtual gridding. The process involves a random assignment of cluster heads to few motes of the SN and then clusters are evolved around them. Representative protocols of this second family of protocols include FLOC [107], EDC [106] and others [103, 104, 109, 110]. A “forced” cluster head is formed in some situations. This occurs when a node is unable to find a cluster head even in a k-hop neighborhood. Simultaneously an optimization process starts for cluster head and member selection for example a node with lowest ID becoming a cluster head. These concepts are discussed in [101, 104 and 110]. The authors of [103] use operations research (OR) techniques to optimize the protocol. The intuition being to minimize 1-hop grids and maximize k-hop wide clusters.

A process of similarity matching amongst nodes is the criterion for cluster formation described in LCNA [100]. This is achieved by first exchanging info about their data readings. In this manner k-hop clusters are formed having similar data. So data
aggregation is achieved with minimum communication overhead. In respect of cluster sizes, the algorithm is a traditional k-hop clustering and because of the similarity of data requirement is random. (see Figure 2.2).

2.2.4 Location and tree based clustering

The shapes and sizes of location and tree based geographic clustering algorithms are well defined due to obvious reasons. A different location based grid building is explained in GROUP[113]. Here the quadrant sizes are tunable. The cluster heads located at grid crossing points, where grids are overlaid on the network. However, cluster head selection raises some issues: several broadcasts are needed for the nodes to converge upon one cluster head in each round and each round needs a wide network broadcast of the next clustering grid center.

The authors in [115] used another geographic based clustering of “multi-resolution”. Using a hash for cluster head locations in the network does this. Thereafter the nearest nodes become part of that cluster head and store aggregated data for further reference. The organization of the network is rather in the form of a tree than cluster-based: when searching for data, the request travels through the tree and aggregated data stored at the vertices are used for routing the request down to the leaf with the required non-aggregated data. Another approach based on tree structure is presented in [114], where first a spanning tree over the whole network is computed. Each node then stores how many children its own sub-tree contains. The protocol traverses all of the nodes and selects some of the available nodes as cluster heads for their corresponding sub-trees.

2.2.5 Infrastructure supported clustering

Some of the clustering approaches assume a pre-existing backbone of powerful nodes throughout the WSN so that the load for the nodes and cluster heads is balanced. Such an approach is taken in [116] where a special metric called “business’ of parent
nodes is introduced. Sensors select their cluster heads such that the load of processing and communication load of all the cluster heads is balanced.

A similar problem is discussed in [118], where the nodes should be associated with a powerful application node such that the overall lifetime of the network can be maximized. Unlike most of the other commonly used clustering approaches, this work assumes that once the nodes are associated with a cluster head, they never change their membership.

2.2.6 Non-uniform clustering in WSNs

Last but not the least significant is the research field of non-uniform data dissemination in WSNs. The basic idea in such case is that the sinks need accurate information from sensors located nearby and less accurate information from nodes, which are very far away. Thus, aggregation needs to be done depending upon the distance to the sinks. The fisheye [122] technique from computer graphics has similar properties, using distance as a measure to determine accuracy. This technique inspired the fisheye state routing [123], a MANET routing protocol in which the nodes exchange routing tables with frequencies dependent upon the distances to the routing table entries. However, the non-uniformity in that case is applied to routing information and not to the data itself.

Similar non-uniform data approaches have been introduced in distributed systems [124, 125]. However, neither of these approaches takes into consideration the energy or CPU processing time and both require global knowledge of the static network, thus making them inappropriate for the WSN domain.

The idea was first conceived for aggregation and clustering in WSN in [121], where a randomized algorithm produces clusters of different sizes depending on the distance to the single base station. In [117], a centralized approach with Voronoi
tessellation or an approximation of it is used to define cluster heads priority, so that the network lifetime is maximized. The cluster sizes grow manifold with increasing distance from the single base station and their shapes and sizes are defined by the location information. However, knowledge of global topology is needed to compute the clustering information.

Another significant work on non uniform or unequal clustering is [105]. It assumes that clusters near to the single base station should be smaller to preserve energy for routing packets from clusters that are more distant. It uses a simple LEACH-like selection scheme; where a random set of nodes compete against one another to become the cluster heads.

Each competing node has its own competing radius, which increases as the distance from the base station increases. At the end of the competition phase, only one cluster head survives in each competing radius and all other nodes adjust their levels of power to reach the closest cluster head. Packets are routed only through cluster heads resulting in quickly draining their batteries. Like any other random clustering protocol, the work in [105] produces random clusters with larger clusters far away from the base station. However, the load is not balanced well and cluster heads drain out their batteries too quickly.

A very similar clustering approach is presented in [95]. However, instead of having random cluster heads competing against one another, nodes exchange their residual energies with all neighbors in their cluster radius. The cluster radius grows manifold as the distance from the base station increases and the node with the maximum residual energy is selected to act as the cluster head. Communication with the base station from any cluster head is direct.
2.2.7 Centralized clustering

In order to compute optimal clusters, many algorithms need complete S-network architecture/and or residual energy information. (e.g. [126, 127]). The entire network has to be updated with cluster information at the end of each round. The basic advantage of such (centralized) protocols is that they can build clusters with any properties. The neither downside being they do scale nor account for fundamental network issues of asymmetric links and failures.

2.2.8 In-cluster data collation and clustering

The primary task of grouping/clustering is to collate many packets and transmit one of the collated/shortened packets. This is achieved by in-network pre-procedures (aggregation or compression) by cluster heads from their group members. Of the many aggregation techniques the authors in [128] have identified three main ones: i) tree aggregation, ii) centralized pre-processing and iii) gossiping. The logic in i) is of good benefit particularly in multi hop clusters. In this (tree aggregation) algorithm the info is processed and aggregated at each hop. The collation is not restricted only to group heads but is distributed over large nodes of the group hence the advantage for multi hop clusters. In centralized pre-processing the data of every group is combined in one central node (CH) and pre-processed at this node-a process akin to LEACH. This combining method has a larger transmission overhead if clusters are multi hop vis-a-vis tree aggregation. Since all the data readings are available centrally, data processing in ii) is more precise. The third aggregation technique (gossiping) is not related to clustering as no clusters are preserved and nodes exchange (gossip) some of their data readings with other nodes for instance to determine the best path from
source to sink.

2.2.9 Optimal clustering analysis

Figure 2.3 illustrates the clusters built by some representative clustering protocols in a sample topology. Looking at them one can easily see the disadvantages or advantages in terms of quantity of nodes per cluster, shape and size of the clusters, etc. However, this is a subjective view and often depends highly on the given particular network. The question about what is the optimal clustering remains unanswered: which is the cluster with a shape and size such that communication overhead for routing data from nodes to base station(s) is minimal?

Instead of evaluating the clustering techniques against standard metrics (in-network processing, communication overhead etc.) some researchers evaluate the techniques in terms of their optimality or hop diameter. The authors in [129] have concluded that the hop diameter of the optimal cluster grows as the size of the network increases. In this work, a simple network scenario is used with one base station, a fixed node density, and a unit disk graph communication model where a node spends energy only when transmitting a packet, and not while receiving it. Another similar work [100] comes to the conclusion that a 2-hop cluster radius is optimal for all practical networks up to 3000 nodes. That multi-hop transmission can also achieve optimal routing has been demonstrated by authors in [100] in certain scenarios.

2.2.10 Clustering in WSNs: Summary
In this section, is presented a part of the wide variety of clustering protocols for WSNs together with some efforts on finding the optimal clustering scenario. Clustering along with data aggregation has been proven to inherently diminish the communication overhead in WSNs, improve delivery rate and save energy. However, there are some major bottlenecks that are not yet efficiently overcome. The first challenge of clustering protocols is the process of selection of cluster heads. In case the cluster heads are pre-known (powerful specialized nodes) their cost and deployment planning is a big disadvantage and the network is hardly scalable. In case the network is homogeneous and any node can serve as a cluster head and aggregator, substantial overhead is needed for agreement on cluster heads.

![Sample clusters](image)

**Figure 2.3.** Sample clusters, as built by some state-of-the-art protocols.

Simple, low-overhead agreement schemes are nevertheless possible, but lead to communication load and highly unbalanced clusters of the nodes.

Another major challenge is the management of mobility and failures of the nodes.
Special failure detection and repair mechanisms are needed to handle these situations, which result in high packet loss, high non-data related communication and processing over-head, and long delay. Last but not the least, a significant part of the clustering approaches rely on only a single base station and cannot be extended to serve more than one of them.

Clustering in this thesis is used to meet the challenges of data dissemination in large networks. Our goal in the designing of the clustering protocols is to solve efficiently all of the above explained problems. Our clustering protocol needs to overcome the communication overhead of cluster head selection. It should be robust against node failures, have the ability to support multiple mobile sinks and also be able to use energy in an efficient and a balanced way. In addition, it should support non-uniform data requirements.

2.3 Machine learning for WSNs

The first intuition for solving the clustering and routing challenges in the application scenario is to apply the techniques of artificial intelligence. In this section, are explored the various available computational intelligence (CI) and machine learning (ML) approaches, which have been successfully implemented to a variety of issues in WSNs. The goal is to better understand their requirements and properties, their application areas, and to identify the best suited approach for our scenario.

The major applications already addressed by CI and ML in WSNs techniques is:
• Sensor Fusion and Data Mining. Is the method of consolidating information received from several supplies so the accumulated data is in some sense higher than the instance once obtained from every source, and therefore note radio powe usage for transmission to the central station cost effective?

This includes computation of data models (data mining), which help the base station or the sensors to differentiate between expected and unexpected data or valid and faulty sensor readings.

• Energy Cognizant (Aware) Routing and Clustering. Best use of power is indispensable in sensor networks, as recharging or substituting the batteries on the nodes may not be cost effective, non-practical or dangerous. In a lot of applications, WSN life span of a few months or years is desired. Now is introduced routing and clustering protocols based on machine learning in addition to those presented in earlier Sections of this Chapter.

• Scheduling and Medium Access Protocols. Sensor nodes are very often restricted by the availability of power and are usually expected to perform unattended for over several months or even years. Thus, it is very important to first identify the largest consumers of power and then to minimize their consumption. It is well known [130, 131] that the primary power consumer on any sensor node is the radio. Thus the MAC protocol becomes the most crucial instrument to minimize energy expenditure in sensor networks. The MAC protocol sits on top of the physical layer and controls the operation of the radio. It manages and schedules its sleeping and idle phases, trying to avoid or minimize collisions, overhearing and idle times. In the latest years,
many efforts have been made to design the ultimate MAC protocol, which minimizes the energy spent for message transmission. A summary of the latest MAC protocols is presented in [131].

- Planning and Operating. WSNs are used in a spectrum of applications from observance of forest fire through airdropped sensors to monitoring a biological system through deep-seated actuators. In few uses, they need to be located at precise and pre-established positions, whilst in others such site awareness is not required or nonsensible. The goal of SN architecture is to determine the kind, quantity and site of measuring devices that need to be placed in a setting in order to get full information of its condition. On the other hand, sensor network deployment copes with hardware and software installation and primary testing.

![Diagram of Machine Learning and Computational Intelligence](image)

Figure 2.4. Taxonomy of Machine Learning and Computational Intelligence, compiled from [132, 133 and 134]

**Localization:** Node localization refers to determining the locations of all sensors deployed in the network. Position knowledge is required to record and detect phenomena, or to guide packets using geographic cognizant routing (see Section 2.1).
Mostly position is the information, which needs to be sensed. A summary of positional techniques for WSNs is discussed in [135].

Some recent surveys give an overview of employment of computation intellect and machine learning for sensor networks [136, 137, 138, 139, 140, 141, and 142]. A general taxonomy of the applied techniques and algorithms is given in Figure 2.4. It is useful to follow this taxonomy to present the individual algorithms and their applications below. However, this study is not aimed to be comprehensive or through. Instead, are summarized the most prominent or relevant efforts for our target scenario and explore their requirements and properties.

### 2.3.1 Neural Networks

An Artificial Neural Network (ANN) is an information processing model inspired by the biological nervous system, like the manner in which brain processes information. It is composed of a large number of highly interconnected elements (neurons) working in unison. Easy routines are usually attributed to each mote (like addition) and weights are allotted to the inter-connections between the nodes.

![Figure 2.5. A generic layered configuration of an ANN with the three layers-input, hidden and output layers. [143].](image)

Massive parallel connections allow data flow through the network using nodes.
and reaching output neurons at infinitesimal speeds. Figure 2.5 gives an example of a simple neural network. The most renowned trait of ANNs is their capability to learn the weights between the neurons are the real computational power and have to be adjusted such that the output is exactly the mapped routine. The process of learning or training of data in ANN is done as follows: The need of training data is met by pre-mapping possible inputs for the required outputs. For instance in a classification problem of hand-written numbers, the images are inputted and classified as numbers for output. Interested readers are referred to [143, 144] for a detailed information on ANN.

**Sensor Fusion and Data Mining**: Neural networks are a feasible solution to many centralized problems like data mining and sensor fusion. The authors in [145] concentrate on the problem of class imbalanced data for sensor based intrusion detection. In their learning protocol, they first gather some real time sensor data; send the data to a given base station, which learns a classification model and sends the model back to the sensor nodes. The main objective is to minimize the communication overhead since the sensors report only positive (intrusion detected) samples to the base station. The approach uses a neural network on the base station to learn the classification model and is fully centralized.

A different data mining problem has been addressed by [146, 147]. In this work, the authors postulate a neural network based approach for checking sensor data integrity or automatic sensor calibration. The key feature of the protocol is the used neural network, a competitive learning NN (CLNN). This NN is an unsupervised learning agent, having the ability to learn data online from a continuous, non-labeled data stream (sensor readings). After the learning phase, the agent is able to differentiate between N different clusters (N is fixed and known before starting the
learning process) and thus has the ability to recognize faulty sensor readings. The authors incorporate the clustering approach in the learning method to minimize the communication cost. Each sensor sends its readings first to a local cluster head, where the CLNN is trained, and then the data is classified, filtered and eventually sent to the base station. The algorithm is semi-distributed, since in theory each sensor node could have its own learning agent. However, the input set is restricted and the learning phase will be very long (only own sensor readings available). The clustered approach taken by the authors is the best way to go, such that a trade-off is found between communication overhead and optimality of learning.

The main objective of [148] is to detect mechanical and biological faults in a sensor monitored greenhouse environment. The authors train two different neural networks to classify mechanical faults (sensor or actuator faults) and biological faults (stressed plants). As an input, they use sensory data from the environment, both historical and current. The data has to be gathered on a centralized sink for processing.

**Application of Neural Networks:**

*A. Routing and Clustering.* Neural networks have been widely applied in WSNs.

- **SIR** is an energy-efficient routing protocol, which assigns a neural network to each node in the network. The nodes make the use of beacons to find out the quality of links to their neighbors and the information is fed into the neural network to learn the quality of the links. Routing is then performed based on a modified Dijkstra shortest path algorithm from a source to a single sink using the learnt link quality. The protocol performs well in comparison to Directed Diffusion [58], but results in a high beacon overhead. Additionally, the implementation of a neural network on each of the nodes has high requirement for memory and might be hard on memory restricted...
sensor hardware.

- **Scheduling and Medium Access Protocols.** A centralized neural network has been applied to solve the optimal TDMA scheduling for a WSN in [150]. However, a centralized computation of schedules does not take into account link and node failures, link asymmetry, mobility etc. Additionally, it incurs high communication overhead in the dissipation of the schedules to the nodes.

**Summary:** It can be concluded that neural networks are an excellent solution for learning network wide data models, which are not liable to change very fast. Examples are models of self-calibration, faulty data, etc. On the other side, both the nature of NNs and the results achieved by the works presented in this section show that they are impractical for distributed tasks like scheduling and routing. Further feasible application areas for neural networks are sink placement, optimal sensor and localization etc.

**2.3.2 Support Vector Machines**

Support vector machines (SVM) are a supervised learning method used for the purpose of classification. The input data is viewed as a set of vectors in an N-dimensional space and the output of the SVM is a separating hyper plane between both sets, which maximizes the difference between each of the sets (the margin between the sets) and the hyper plane. For computing this hyper plane two parallel hyper planes are constructed on each side of the separating one and “pushed against” the data sets. Samples on the margin are called the support vectors.
Figure 2.6. Separating hyper plane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Source: www.wikipedia.org

Figure 2.6 presents a simple example with two data sets (classes) in a 2-dimensional space. The two parallel hyper planes on each side of the separating hyper plane together with the data samples that they include are called the support vectors.

Localization: A solution to the localization problem with support vector machines has been proposed in [151]. Given n + m motes in the SNW, where the positions of
motes are unknown and of \( n \) nodes are known, and given the RSSI signal strength between any given pair of nodes, the positions of the un-localized nodes have to be recovered. The authors first train a SVM for classification of nodes depending on their distance with each other. It is then required to match the output of the SVM to the positions of the nodes. The algorithm used is fully centralized, which is a consequence of using a SVM. Other researchers have also used SVMs for the purpose of localization in WSNs [152, 153].

**Summary:** Just like the other supervised learning approaches, support vector machines are memory processing, intensive and need centralized gathering of the input data. They are suitable for solving data mining problems like sensor fusion. Additionally, they are also suitable for localization, since it is usually done only once right after deployment. Other centralized problems like optimal sensor placement are further possible applications.

### 2.3.3 Decision trees and case based reasoning

These two similar techniques are based on the idea of classifying items into even smaller clusters. It is like classifying an orange first as fruit, then as a citrus fruit, then as an orange. These data mining algorithms are relatively fast to train, very easy to understand and very fast to execute. They require that the items, which are needed to be classified, are attribute-value pairs. For example, an orange can have attribute-value pairs color = orange, shape = sphere. In decision trees there two main rules for creating the decision tree: ID3 and then its next C4.5 [134]. Basically, they need to answer the question "which attribute to check at the root of the tree, which next?" A formal description can be found in [134].
**Energy Aware Routing and Clustering:** An application to link quality classification in WSNs is presented in [154]. The author’s simple rules for classification of links into good and bad, based on the buffer sizes, RSSI level of packets received, etc. The computation is done centrally on the base station and the data model is disseminated to all nodes in the wireless sensor network.

**Summary:** Decision trees and case-based reasoning are practically feasible techniques for small size localized problems on individual sensor nodes or larger data sets on a centralized base station. They are simple to deploy and implement, but do not lead to optimal results.

**2.3.4 Reinforcement learning**

Reinforcement learning (RL) [155, 179] is interested with how software agents should take actions in a setting so as to optimize some odds of additive reward. Thus, it elects some attainable course and gets a prize from the setting for the requisite action or course taken. Observe that the most effective choice at some point is rarely identified beforehand. Consequently, the agent must strive many alternative actions and sequences of actions and eventually learn from its experiences. A popular instance is a mouse or a robot learning to move in a maze environment (Figure 2.7). At every step it can select one action from a pool of available actions based on its running state of the scenario and previously acquired information. It completes the move and gains a prize from the environment. Usually the reward is negative when the goal is not reached yet (e.g. the cheese is not found) or positive when the goal is reached.
The authors point out that RL is compatible for distributed issues, like routing. It has moderate needs for memory and comparatively low computations are needed at the particular motes. This happens as a result of the necessity of keeping many various potential actions and their information. Therefore it desires a while to converge, however it’s versatility to topology changes, simplicity to implement achieves optimal results. The most widely used reinforcement learning algorithm is Q-Learning, which assigns a Q-Value to every possible action representing their goodness or quality. After learning, the best Q-Values mirror the optimal actions.

**Energy Aware Routing and Clustering:**

Q-Routing is one of the early and essential approaches of packet transmission [157]. The researchers explain a really easy, Q-Learning founded algorithmic program that exploits the most effective ways considering minimum latency to the destinations.
Possible actions are next hops at the nodes, and a Q-price is assigned to every combine (sink, neighbor) representing the time that a packet wants through this neighbor to reach the sink.

Simulations have shown the algorithm to be economical once the network masses are fairly high. It also performs well under changing topologies in the WSN. Whilst the procedure was manifested for the wired and packet-switched networks, it inspired a lot of work in the WSN and wireless ad hoc communities, because it is fully distributed. A recent implementation on CrossBow motes [158] has demonstrated the practicality.

Many other routing protocols have been inspired from Q-Routing [159, 160, 161, 162, 92, 163, 164, and 165]. The main difference between them is the cost metric, which is used for routing. Delivery time is used in [166, 167], maximum compression paths are learnt in [160, 161, 164], and geographic-based routing is implemented in [159, 163]. A novel cost metric is used by [92], where the routing protocol learns to avoid "important" nodes: nodes, which after failing might disconnect the entire network. Neighboring nodes exchange information about their importance (computed locally at the nodes based on full topology information) and the best routes (with least important nodes on them) are learnt. A more general cost function is defined in [165], where any combination of number of hops, delay, and remaining energy on the nodes can be applied.

Another difference between the above approaches is the reinforcement learning algorithm used. The authors of [166] use dual reinforcement learning, which gives rewards not only for previous actions, but also for next ones. The convergence of learning is quicker and the protocol shows improved behaviour. Q Learning is used by
A reinforcement technique for robotic soccer-Team Partitioned Opaque Transition (TPOT) is developed [168] and applied to packet routing [167]. It permits a body of freelance knowledge vehicles to cooperatively acquire a shared work, like taking part in football. It varies from ancient RL in its worth operate, that is divided among the agents and every agent acquires solely the half that is significant directly to its localized moves. Also, the setting is hermetic to the agents, which suggests that they need no data concerning their goodness or next doable actions of their mates.

Reinforcement Learning (RL) is learning by interacting with the environment [169]. It studies a rft lrng rule, created for cracking the point-to-point rtg prbm in MANETs. Collab RL (CRL) is strongly supported on Q-Learning employs additionally a decay fn (similar to pheromone evaporation in ACO, see further Section 2.3.5) to better meet the requirements of ad-hoc networks. An additional contribution of [161] besides the Q-Learning routing protocol is that the automatic learning of optimum parameters of the algorithmic rule with a theorem (Bayesian) exploration strategy. Certain algorithms require parameter pre-setting before successful use. Application of this algorithm to RL or non-RL based models is presented in the paper.

The setting of [170] is comparable to those above: one base station is receiving data from several sources. The rule reflexes under consideration the remaining power on the nodes, the aggregation magnitude relation, the link reliability between the nodes and therefore the hop price to the base station. The software agents are once more the motes and Q-worth is allocated to every potential subsequent link at every node. Throughout every occurrence, the present Q-worth is used for routing a message to
the central stn. At every hop, the complete hp data is attached to the message (balance engy, rwds, etc.). Premiums are determined at the bs. Once sufficient pkts are accumulated at the BS, it calcs the Q-Values offline for the motes in the network and distr them across the NW thru a broadcast. To conserve processing energy the base station calculates the Q-values off-line when it has enough such packets.

Although all of the above studies show encouraging results from applying various reinforcement learning algorithms to routing in WSNs, none of them has reached the state of a mature communication protocol with evaluation and implementation in a realistic simulation and real hardware environment. Their evaluations are rather preliminary and concentrate on only a few of their properties, leaving out important questions about overhead and efficient implementation.

**Scheduling and Medium Access Protocols.** Actor Critic Algorithm [171] is an early reinforcement learning algorithm, where the policy is detached from the learnt action values. In current RL algorithm like Q-Learning the policy is fully dependent on the learnt Q-Values, which represent the current state of the value function. This incurs search overhead when the best Q-Value needs to be computed. In actor critic algorithm a separate table (called the actor) can be defined together with the value table (called the critic) to speed up selection action. This algorithm has been applied, for example in point-to-point communication in sensor networks [172]. Concretely the rule optimizes the throughput and hence the energy dissipation in the network. The modulation and power levels are decided by tracking the ctc bfr sz & lst xmission increase. The authors have tested the algorithm on two as well as multi-
mote scenarios. However a comparative analysis has not been done to determine the
gain of the RL algorithm over others.

RL-MAC [173] is a novel adaptive Media Access Control (MAC) protocol
application of reinforcement learning to adjust the schedule of sleep and active duty
cycles in a WSN setting. Traditional schemes center on scheduling this cycle to
minimize energy consumption. The duty cycle in this algorithm instead of being
constant is made variable based in the current traffic. A node transmits control
knowledge including a reward for other nodes at the beginning of its reserved slot.
The parameters of waiting messages and those sent successfully in the reserved slot
determine the reward function of that node. The rwd fn is based on the
No of successfully transmitted msgs during the reserved slot and the No
of waiting msgs on the nodes. In this way the nodes actively infer the state of
other nodes using a RL based adjustment mechanism, thereby achieving high
throughput and low power consumption for a wide range of traffic conditions.

The algorithmic rule of COORD discussed in [174] is a distribution RL to realize
the best coverage. Tracking the active and sleeping motes in the network till the full
sensing of the phenomena achieve this. A 3-way approach supported by Q-learning is
presented. The desired actions are i) transmitting from active to sleeping ii) and back.
The entire network is split into oblong-groups and every grid vertex needs to be
covered by at least one mote. The number of nodes to a vertex is the Q-value of that
vertex. The Q-value table of all the vertices is updated after each run. An associated
action is done based on the Q-value. Subsequently, the next run commences and Q.
tables are updated & soon. Comparable results are found of the other two solutions
The Q-learning approach used by the authors’ indicates a good model of converting a controlled problem into a distributed one. Some drawbacks—a clear protocol implementation is missing as also the coordination and states of the grid vertices. The approach can be run on line if needed as it is fully distributed.

**Service Delivery. Design and Deployment Issues:** The authors in [175] use a discovery protocol. Making the problem a Semi-Markov Decision Process solved by the Q-learning technique of R L optimizes the network-topology problem. Though the protocol is designed for MANETs, it can also be successfully applied to WSNS for localization.

**Summary.** Reinforcement learning is one of the most widely used ML techniques for distributed problems in MANETs and WSNS such as scheduling, medium access control, routing, service positioning etc. Its most important strengths are the online learning algorithm and the model-free nature, but also its flexibility and fast adaptability to changing environments. RL implementations for WSNS incur only minimal communication overhead and achieve optimal results. Thus, RL should be the first choice when solving distributed problems in WSNS.

**2.3.5 Swarm Intelligence**

The management of WSNSs by various techniques for routing and localizations has been highlighted. To cope with the increasing size of such networks, the authors in
[176] developed a new class of algorithms motivated by swarm intelligence. Natural biological swarms have numerous powerful properties desirable in communication and sensor networks. Just as ants use the shared trail of ‘Pheromone’ software agents can use similar shared memory and computational resources. A more general overview of Swarm Intelligence can be found in [177].

There is two main branches of swarm intelligence: particle swarm optimization (PSO) and ant colony optimization (ACO). The first technique was developed by Kennedy and Eberhard [177] in 1995 and is inspired by bird flocking or fish schooling. It is applicable to problems where the solution can be represented as a point in a search space. Agents are points in the solution space and possess movement speed and direction. Usually a high number of agents is used to represent many different solutions. During learning, agents move around in the solution space and are evaluated at each step according to some fitness function. With time, individual agents accelerate towards other agents with higher fitness in their direct neighborhood, thus forming schools or flocks. Figure 2.8 illustrates the main concept of PSO. The algorithm is extremely resilient to the local minimum problem, because of the high number of agents.
Figure 2.8. Particle swarm intelligence (PSO) in action: particles are initialized at random positions (top) and after learning cluster into groups (bottom) [178]

Ant colony optimization, the second technique was presented in [180] by Marco Dorigo for the first time. The basic idea is of graph optimization. Ants leaving pheromones for future ants to follow traverse the edges. A comprehensive description of ACO is also given in [181]. (Figure 2.9 illustrates this technique)
Figure 2.9. The double bridge experiment for finding shortest paths with Ant Colony Optimization. (a) In the beginning of the experiment ants take explore all possible routes. (b) At the end of the experiment most of the ants take the shortest path to the foraging area, while few ants explore other non-optimal routes. Copyright [179].

**Energy Aware Routing and Clustering:** It is seen that routing is the process of guiding information from source to sink and clustering a method of or a subset of routing. In [182] four types of Particle Swarm Optimization [PSO] for energy aware clustering are analyzed. The basic concepts of the algorithms can be summarized. Artificial ants are asynchronously launched towards the destination, at regular intervals. Each artificial ant searches for a minimum path joining the source and Base station. At each intermediate node a greedy stochastic policy is applied to choose the next node. During backward travel, the pheromones stored are modified as a function of goodness of the path. The rule is predicated on a straightforward concept. Cluster
heads of various clusters are selected based on location. The nodes closest to the base-station are selected as cluster heads. A basic deficiency of this rule is that the clusters and CHs depends on distribution of the motes and is centralized. Thus, in case of failures or any topology changes, the new information needs to be gathered at the base station and clustering needs to be recomputed.

A novel agglomeration or clustering approach for WSNs referred to as CRAWL is outlined in [183] with the utilization of soldier ants is again a probabilistic formulation to finding good paths in a graph that is a network. In this case an alternative ant exists for the main one making the algorithm more robust. A process is in place to select cluster heads. Each node computes its avidity value periodically which is broadcast. The node with the lowest value is elected as the CH. The cluster head floods the network with this new clustering knowledge so that nodes adjust themselves and use new power levels (lower) to transmit to the CH. The logic assures that CHs are chosen based on adequate energy reservoir and sturdy neighbors. At times the packets may flow away from the sink as an intermediate node has a higher value, making the rules suboptimal for minimizing the energy of individual motes, but is good in terms of constructing effective energy utilization of the entire network.

AntNet [184] is an ACO application in communication networks used to find near-optimal routes in a communication graph without global information. The agents are divided into forward and backward ants. Forward ants are initialized at the data source and sent to all known destinations at regular intervals. They travel through the network graph by randomly choosing the next hop and leave pheromones on their way. The more ants have chs the similar pth, the hghr the phrmon lvl of that path. Drng their tvl, fwd ants collect rtg info, indicating the
arr time at every mote on thr highway. At destn arr, the fwd ants are transformed into bckwd ants and use the cached rt they have tvl to tvse the same route again and to updt the phrmne tab accrdg to the collected rtg info. Details of this computation can be found in [184, 48]. A decay function is implemented as evaporation of the pheromone levels, indicating which routes are the most freshly used ones. The version of AntNet for MANETs is called AntHocNet [48] and is developed by the some of the authors of AntNet. AntNet and AntHocNet use both reactive path setup and proactive path maintenance for single source -single sink. However, the approach requires ants to be traveling independent from data packets and even to trace each path twice (forward and backward), which causes a great overhead and is not well suited for energy- restricted WSNs. Nevertheless, the method is fully distributed and is the one bst expl & described in the lit for using swrm information in wrls NWs.

AntNet [184] is associate ACO application in communication networks wont to realize close to best routes in an exceedingly communication graph while not world info. The agents’ square measure dvdd into fwd and bcwd ants. Fwd ants’ square measure initld at the information supply and dispatched to all or any knwn destns at rglr intrvlsls. They trvl thru the NW grph by indiscriminately selecting ensuing hop and leave pheromones on their way. Then a lot of ants have chosen an equivalent path the upper the pheromone level of that path. Throughout thr tvl, fwd ants collect rtg info, indicating the time of arrival at every mote on their method. At destn arr, the fwd ants square measure remodeled into bckwd ants and use the paid rt they need tvl to tvrse an equivalent route once more and to updt the phrmn tab in step with the collected
Routes are most freshly used can be de-alienated using a decay function of the evaporation pheromone type. The version of AntNet for MANETs is called AntHocNet [48] and is developed by the some of the authors of AntNet. AntNet and AntHocNet use both reactive path setup and proactive path maintenance for single source -single sink. However, the approach requires ants to be traveling independent from data packets and even to trace each path twice (forward and backward), which causes a great overhead and is not well suited for energy-restricted WSNs. Nevertheless, the method is fully distributed and Ant Hoc net is the most researched technique of using swarm knowledge.

Another swarm intelligence based protocols for MANETs is discussed in [49] and is called Multi cast for Ad Hoc Networks with Swarm Intelligence (MANSI). In a typical multicast protocol the core node through ‘Jn Req Pkt’ and a bckwd ‘Jn Rpy Pkt’ initiates the growth of a multicast tree. In MANSI, nodes that are not core send ants into the network to find out good rts to the core, lve the guidance data (pheromones) in the process. Following ants to select their hops subsequently uses this knowledge. The optimization is multi cast and hence differs from Ant Hoc net [48], which is unicast. In [185], the authors propose associate degree AntHocNet [48]. This is useful for sensor networks in buildings. That returning ants create an overhead is the basic drawback of this rule.

The cost of sending ants is so high in Ant-Based control (ABC) [186] though a lot of features are common to Ant Net. Another difference of the protocol is that only one
cl of ants are sent at reg interval from the information source moving stochastically through the network and updgt the rtg info as they travel to their destination. Once at destn they are destroyed. Thus only the forward ants have a smaller communication overhead and so ABC is well suited for WSN setting, albeit the drawback highlighted earlier.

The delivery of fresh or new info is done by what are known as ‘mobile agents’ a term, which causes ambiguity for machine learning or swarm intelligence. Essentially they are simple, small entities (that is packets), which travel through the sensor network delivering updated information to the notes. The agents in case of routing update the path or next hops on nodes. [188,189 and 190] Whilst this mechanism is very efficient in some application they cannot be classified as learning or as a swarm intelligence algorithm. This type of process is used in Uniform Ants [187] and is a smpl ant optmzn keyed tech. It optimizes and maintains routes in a MANET. The next hop is selected on a ‘greedy’ or uniform basis. It needs to be noted that, they also increase the overhead cost for sending the agents.

**Design and Deployment:** PSO has various applications to design and deployment in WSNs. It has been successfully applied to optimal detection coverage in maritime surveillance in [191], to finding optimal sink paths across a sensor field [192] and topological planning for traffic surveillance in [193]. All of the applications use the original PSO algorithm, with different parameters for the particles’ speed and acceleration.

**Localization:** A suitable application area of PSO is also localization in sensor
networks. In [194], the base station runs a PSO-based algorithm with centralized information to find the positions of the network nodes. However, PSO is a distributed technique and can be applied also here as such.

**Summary.** MANETs are Ad Hoc and distributed networks, where energy is not limited. Swarm intelligence is ideal for such network where topology changes are paramount. Interestingly, PSO has been applied only in a centralized manner, although it is a distributed technique and network nodes could represent individual particles. ACO, on the other hand, has been applied mostly to routing and has proved to be an efficient and flexible algorithm. In the context of energy-restricted WSNs, PSO seems the better choice because of its localized nature and small communication overhead. To the best of our knowledge, there are no PSO applications to routing in WSNs. All WSNs applications of ACO suffer from the great communication overhead of the traveling ants. However, a different implementation of ACO is also possible, where ants carry data packets and thus minimize exploration overhead.

### 2.3.6 Genetic Algorithms

The paradigm of genetic algorithms (GA) is based on biological evolution. It describes a system, consisting of individuals (chromosomes, genes), which evolve through cross-over (combination of two individuals) and mutation (spontaneous change of the properties of one individual). The individuals are organized into generations and represent possible solutions to the problem: with time, the properties of the generations change and evolve and the solutions become better in terms of some predefined fitness function. The general model of genetic algorithms is
illustrated in Figure 2.10. More information about genetic algorithms can be found for example in [195].

![General model of genetic algorithms](image)

Figure 2.10. General model of genetic algorithms.

Genetic algorithms are easy to understand and the system easy and fast to define. However, they require centralized computation and converge slowly. Since they keep at least two full generations at any time to be able to compute the next one, they have also high memory requirements. However, their biggest disadvantage is their inflexibility in case of changes of the input: the whole evolution process has to be rerun in order to find a new solution.

**Sensor Fusion and Data Mining.** The implementation of genetic algorithm (GA) for info aggr in a tgt det setting is discussed in [204] using mob agents.

Selectively the agent browses the motes and incrementally fuses the activity data. On comparison, routes determined by GA are superior to those determined by more used “local nearest first” (LCF) and “global nearest first” (GCF) algorithms. The expenditure in energy terms to assemble info to a control limit of the optimal path is not calculated. An extension of the study in [196] is presented in [197]. In addition to data acquisition and processing time, this study also includes agent transmission time delay in a route R in the fitness function definition.
Energy Aware Routing and Clustering. The ‘First Note Dies’ (FND) an important network longevity parameter is analyzed in [198], using genetic algorithm. The methodology is named GA-routing. It creates summation trees that scale all the notes. The continuous use of any path would cause early failure of notes in live. So whilst this algorithm creates optimized paths, this drawback has to be addressed. The approach in [198] tries to find a solution by determining to longevity of a selected tree before the next tree becomes operative. The spanning trees are sculptured as individuals. Simulation results show that GA provides higher time period than the one best tree (SBT) rule, and also the same network life as the cluster based mostly most time period information aggregation algorithmic rule [199] for little network sizes. However, the algorithm's overhead is not evaluated.

Representation of sensor notes as pieces of chrmn-clstr heals as one & member as O. Is another GA based algorithm for optimizing clustering and is presented in [200]. The number of motes equals the bits of a chromosome. The strength of chromosome is totalized. A higher energy conservation has been exhibited by GA approaches in comparison to similar algorithms [201]. An often used GA algorithm is that of two phase localization – use a combination of GA and simulated annealing. The algorithm address a prbm clld ‘flip ambgy’. It is possible tht the dist related loclzrn may not be unique, when more than that one note is on a collinear (straight line).This problem is comprehensively discussed in [202,203 and 204].

Scheduling and Medium Access Protocols. An animated sleep scheduling model postulated on GA is presented in [205]. This model works well for large randomly developed WSNs. Here, the networks deploy a large number of redundant motes for
bettr cvrg. Obviously combining motes for a prlngd NW life is a maj prblm.

If a task is to be performed in a given geographic area the network is divided into active and sleep partitions, with only those motes active that cover the phenomena and have adequate connectivity. If some nodes die, causing a blind spot (s), then entire network is woken up to take a fresh decision. This is clearly a dual objective (A subset of multi objective) optimization problem – to minimize energy consumption and reduce the number of active nodes.

The study in [206] proposes a Decision Support System (DSS) to establish the best sensor architecture. In this scenario, nodes are partitioned into clusters with local cluster heads, which dictate active intervals to the nodes. Active intervals need to be coordinated among clusters to avoid intra-cluster interference and minimized to minimize energy expenditure. Again, the proposed algorithm is centralized and does not take into account crucial WSN properties such as failures.

**Design and Deployment.** A decision support system (DSS) based on GAs is proposed in [207]. The DSS determines the best WSN design, which is to be used by process engineers. Usually the engineer first defines some measurable quality metrics, selects an initial sensor network design and evaluates it. Depending on the achieved results, she changes the design and re-evaluates it. The DSS presented in [207] automates this process by feeding random network designs into a GA and searching for the best solution according to the defined quality metrics. On one side, this is a valuable tool for WSN designers and speeds up their work. On the other side, their expertise is still crucial, since they need to define the quality metrics and to define how the optimal solution looks like.
Localization. The issue of localization is best resolved through GA approach even though they are centralized [208]. As the problem involves high variance in the data, due to Received SSI val to calculate dist btwn motes, lrge dta sets are required to localize the rslts into least error. The approach assumes anchor and non-anchor nodes to implement localization. The algorithm assumes the full distance information is available on a centralized base station. A similar technique with slightly different fitness functions is used in [209, 210].

Summary. Genetic algorithms have high memory and processing requirements and are very inflexible in case of an environmental change. Nevertheless, they can be used for some centralized problems, where the results need to be disseminated only infrequently to the nodes. Examples are localization in mostly static networks or sensor network design and optimal positioning.

2.3.7 Heuristic Approach

Experience based srch mtds work in 2 stages: Plng & exec or implementation. For instance wkg with a discovery tree. First it calcs the cost fn of all motes and then transverse the bst psbl route thru the tree. Such exhaustive search techniques are impractical in real world problems, as agents to perform the search and decisions have to be made conditional to availability of local info only.
RI time learning srch mthds termed agt-cntrd op well in such envrmnts [211].

The belief it can achieve in the subsequent advance are determined by the agent based on the current neighborhood status and implement its next action such as a next hop. Figure 2.11 illustrates the general model. A simple example is a robot, trying to find its way in an environment full of obstacles and to reach some goal position. It will evaluate its immediate action possibilities (movements) and choose the best one. After this planning/execution step, the robot will re-evaluate its current state and so on. Crucial for the algorithm is the evaluation of the current options of the learning agent. They need to be initialized with a globally known fitness function. Learning in Real Time (LRTA) is an algorithm as an outcome of such localized discovery and action [211, 212].

**Energy Aware Routing and Clustering.** A popular model to evaluate search of a neighborhood has the following intuition: the nodes in the network can be modeled as the agent states; the packets as the agents and the information available at the nodes about their one-hop neighbors. This is a typical RT hrstc srch method amply suited for wrls ad-hoc settings. LRTA* is analyzed to routing in ad-hoc networks in [213, 214] with good results. However, the need of a global heuristic limits the applicability of the algorithm in distributed environments.

**Summary.** Here (that is in real time heuristic search) the maiden values of the states are estimated as pth csts to the gl (destination). If the learning is acceptable that is pledged never to over-estimate the real cost, an optimal solution is found by the algorithm. It might appear that RT hrstc srch is akin to the Q-learning but that is not the case. Heuristic searches require gbl knwlge of the surroundings and no expln of non-optm rts is done. In the presence of such a heuristic,
like available location information for the neighbors and the sinks, the approach is feasible. On the other hand, reinforcement learning is a better choice because of its ability to learn from previous experience.

### 2.3.8 Fuzzy Logic

Intelligent or AI techniques find various applications in wireless (sensor) networks. The advantage of fuzzy logic is that it is capable of real time information even with less information. Conventional systems rely on a precise representation of the scenario, which in real world setting does not exist. On the other hand fuzzy logic systems can maneuver the routes in a natural way and so are suitable in this regard. Also it can be used for context by mixing different parameters-rules combined to produce suitable results. Different aspects of fuzzy logic are described in [215]. The logic is different from set mathematics where elements are enclosed or not. This is also different from the human measure of inexactness or uncertainty that use variables like most, many frequently and so on. Thus in case of fuzzy logic an individual could be a member of a tall set. Or, as Figure 2.12 shows, the current temperature of a room can be 0.6 cold and 0.4 warm at the same time.

![Figure 2.12. Fuzzy logic example. The classification of some variable (temperature) is not binary like cold OR warm, but fuzzy like a little bit cold and little bit warm.](image)

---

*Figure 2.12. Fuzzy logic example. The classification of some variable (temperature) is not binary like cold OR warm, but fuzzy like a little bit cold and little bit warm.*
In general, Fuzzy rules are of the type that if ‘something happens’ then ‘the consequences are’. Or continuing our example from Figure 2.12: IF temperature is cold, THEN turn on the heating. It is important to note that fuzzy rules contain only IF statements and no ELSE statements. Each of the rules is evaluated individually and independently from each other, since any of them (or all of them) can be true. The ‘IF’ and “THEN” rule of a fuzzy kind of create the fuzzy input house and fuzzy output house respectively. Non-fuzzy inputs (e.g. this temperature) are mapped to their fuzzy illustration (e.g. cold, warm, and hot) within the method referred to as fuzzification. This logic has been applied successfully up to speed sys (e.g., management of automotive theme, pwr sys, home applns, elvtrs etc.), digital image method and pattern recognition.

**Energy Aware Routing and Clustering.** The study of [216] uses three fuzzy descriptors to elect cluster heads. The outer Base station does the selection using the three descriptors – Node connection, Energy Level and Centrality of that node or in the context of the entire network. The study opines that a central algorithm controlled by the base station produce the better cluster heads since the base station is global info about network and is more powerful than the sensor nodes and has adequate memory, power and storage. In this technique, energy spent to send the location information, possibly using a GPS receiver of all nodes to the base station. Since normally WSNs are deployed over a geographical entity with a primary aim of sensing and accumulating info, it is reasonable to assume that the nodes have minimal mobility. Thus, locating can be sent during the initial setup phase. The process is similar to LEACH, but the base station rather than being distributed does the phases.

Intelligent cluster leader election will cut back the power dissipation and lengthen the period of time of the WSN. A fuzzy reasoning technique supported remaining
energy and site info is planned for clstr leaders selection [217]. The report establishes a NW architecture in that all detector motes send the knowledge regarding their position and accessible power to the central place. The central stn accounts under consideration the en every nde has, the amount of motes within the neighborhood, and a mote's dist fm different ndes and determns that which motes ought to function as clstr hds. The central place fuzzfys the var mote en and mote conc into 3 lvls: lo, med and hi, & therefore the var mote distance from base station into close, adequate and far.

The fzy o/come tht rep the likelihood of mote bg chsn as a clstr hd, is split into 7-lvls: terribly tiny, sml, rthr tiny, med, rthr giant, lrge, and really giant. The art obs substl inc in NW alive time as compared to a NW tht u the lo en adpt clstr hirchy (LEACH) appch. However, the approach is centralized and incurs substantial overhead for collecting necessary information at the base station and disseminating the cluster head roles.

**Scheduling and Medium Access Protocols.** [218] presents a fuzzy logic path for this function that caters for confident media access control as also guard against collision, unfairness attacks and exhaustion attack. The perception here is to choose a membership value II evaluated on the basis of clustered density. In this fuzzy logic the sensor compactness can be defined as Very High Density Cluster (VHDC), High Density Cluster (HDC), Average Density Cluster (ADC), Low Density Cluster (LDC), and Very Low Density Cluster (VLDC). The maximum number of motes or nodes in a cluster can be defined by ‘n’. In the classical S-MAC protocol, the wake and sleep times are the same. But in the fuzzy logic system of the study the values are
set between 0 and 1. For illustration assume n=10, and also the maximum number of
motes in the cluster as 10 then it is a VHDC and so the fuzzy membership value is set
to 0.0. 0 that implies the significant difference between sleep and wake times.
Similarly one defines HDC to 0.3 and so on till VLDC is 1. In VLDC, the cluster
contains only one sensor, so the sleep and wake times are the same. Thus, based on
the membership value of the clusters, the sleep and wake times are adjusted.

Therefore, the data or control transmissions of normal nodes fail due to collisions.
Unfairness attackers send attack packets to the common channel immediately on
sensing the channel as idle. Unfairness attackers have a higher chance to access
common channels than other normal nodes Exhaustion attacks occur when adversaries
send great amount of RTS dissipating their power quickly. These variables are
represented as fuzzy and the output is again a fuzzy variable representing the
probability that an attack was detected. The node stops sending/receiving packets
when an attack is detected and goes to sleep for some period of time. After that, the
medium state is re-evaluated. Performance of FSMAC is compared to popular
techniques and results of FSMAC out-perform. The fuzzy model needs to be
disseminated to all nodes in the network.

However, it is not expected to change often and the medium evaluation is performed
in a distributed manner. On the other side, the extension of the network lifetime is
probably due to the enforced sleep mode during an attack.

**Summary:** Fuzzy logic is well suited for defining and solving complex multi-
objective functions. Examples are congestion control, attack discovery, and optimal
sensor deployment. The main challenge lies in defining the fuzzy variables and
determining the fuzzy rules. Usually this needs to be done offline and manually, and then the fuzzy model to be disseminated to the network nodes. However, this is feasible for problems whose models are not expected to change fast -like the above examples.

2.3.9 Summary of Machine Learning and Computational Intelligence techniques

There are many applications of various machine learning and computational intelligence techniques to WSNs. The main goal of this survey and classification is to compare the suitability and applicability of the different ML approaches to the main topic of this dissertation: routing and clustering. Figure 2.13 summarizes the presented works. The suitability of the different ML and CI approaches is evaluated and the resulting protocols and algorithms for WSNs are cited.

<table>
<thead>
<tr>
<th>Application in WSN's ML approach</th>
<th>Sensor Fusion/ Data Mining</th>
<th>Routing and Clustering</th>
<th>Scheduling and MAC</th>
<th>Design and Deployment</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Networks</td>
<td>[146], [147], [148], [145]</td>
<td>[149]</td>
<td>[150]</td>
<td></td>
<td>[151], [152], [153]</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td></td>
<td>[154]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Trees</td>
<td></td>
<td></td>
<td>[153]</td>
<td>[172], [174]</td>
<td>[175]</td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td>[159], [160], [157], [158], [161], [162], [167], [170], [164], [165]</td>
<td>[173], [172], [174]</td>
<td>[175]</td>
<td>[172], [174]</td>
<td>[175]</td>
</tr>
<tr>
<td>Swarm Intelligence</td>
<td>[183], [184], [48], [182], [185], [49], [187]</td>
<td>[193], [192], [191]</td>
<td>[194]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>[196], [197]</td>
<td>[199], [202], [200], [198], [203], [204]</td>
<td>[206], [205]</td>
<td>[207]</td>
<td>[209], [208], [210]</td>
</tr>
<tr>
<td>Heuristic Approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>[213], [214]</td>
<td>[217], [216]</td>
<td>[218]</td>
<td></td>
<td>[219]</td>
</tr>
</tbody>
</table>

Figure 2.13. Summary of ML and CI applications to WSNs. The suitability of the algorithms to each of the surveyed applications in WSNs is shown, together with the surveyed works.
Concerning routing and clustering in WSNs, it can be concluded that there are four well suited ML and CI techniques and one less suited approach. In general, all of the suited techniques are distributed, simple to implement and have little to medium processing and memory requirements. While Figure 2.13 concentrates on the general applicability of the proposed algorithms, Table 2.1 goes one step further and compares the most suitable of them in terms of their properties: memory and processing requirements, optimality, and flexibility in case of failures.

Fuzzy logic (last approach in Table 2.1) has higher computational requirements because of the offline fuzzification of the objective function. Additionally, the fuzzy rules have to be stored at all nodes and their number grows exponentially with the number of fuzzy values of each of the variables. The dissemination of the fuzzy rules is responsible for the incurred additional communication overhead. The results achieved by fuzzy logic are near-optimal because of the fuzzification process - the exact optimal solution is hard to find. Additionally, in case of changes of the objective function, the fuzzy rules need to be recomputed.

<table>
<thead>
<tr>
<th>ML/CI Approach</th>
<th>Comput. requirements</th>
<th>Memory requirements</th>
<th>Flexibility</th>
<th>Optimality</th>
<th>Add. overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reinforcement learning</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>Optimal</td>
<td>low</td>
</tr>
<tr>
<td>Swarm intelligence</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>Optimal</td>
<td>high</td>
</tr>
<tr>
<td>Heuristic search</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Optimal</td>
<td>medium</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Near optimal</td>
<td>medium</td>
</tr>
</tbody>
</table>

\(\text{Table 2.1. Properties of basic Computational Intelligence Paradigms}\)
Heuristic search is very similar in its properties to reinforcement learning. However, it requires a globally known heuristic function, which increases the incurred communication overhead. Assuming the heuristic is admissible, the achieved results are optimal.

Swarm intelligence is a widely used technique for routing in MANETs, where it performs very well under high mobility scenarios. Usually Ant Colony Optimization is used. However, the traveling ants incur high communication overhead throughout the network lifetime.

Reinforcement learning, on the other hand, seems to be the best performing and suitable technique to apply to routing and clustering in WSNs. It achieves optimal results at low processing and medium memory costs and is highly flexible in case of failures or topology changes. The incurred additional communication overhead is minimal.

2.4 Concluding remarks

This chapter presented an extensive survey of state of the art work in routing and clustering for wireless sensor networks and applications of machine learning to various problems in WSNs. A lot of research effort has been invested in these topics, but most of the work presented here suffers from some restrictions. Often the routing or clustering protocol is implemented for a very specific application scenario and cannot be easily applied to other scenarios. Many of the algorithms cannot cope efficiently with node and link failures or mobile sinks. Especially clustering protocols
incur a lot of communication overhead for agreeing on the network structure. Last but not least, machine learning based approaches present a theoretically well designed solution, but do not implement a real-world communication protocol nor do they evaluate or compare it to traditional existing ones.