Chapter 2

Literature Review

In this chapter review of relevant work in the area of cross layer protocols and architectures for Wireless Sensor Networks (WSN) that serves as a background to the research are discussed.

2.1 Cross Layer Design

The traditional approach to communication protocol organization is to use layering wherein, individual protocols are piled on top of each other. Each layer uses functions of layer directly below it. The benefits of layering are: it is easy to manage the protocol stack; it reduces design complexity; offers flexibility; and promotes modularity and software reuse [Tanenbaum and Wetherall].

Due to extreme scarcity of computational, memory, and power resources in the nodes, WSN prefer protocols that require minimal processing, memory, and energy for its operation. This requires architecture to avoid duplication of work at different layers by richer interaction among the layers. Above all, WSN applications require to maintain transmission reliability over noisy communication links. To meet these requirements, network performance is to be emphasized more than modularity. Further, resource scarcity issues are to be handled at different layers in a profoundly knotted manner. For example, information of strength of signal received from a neighbouring node (physical layer information) can assist routing protocols (at network layer) to decide next hop in the route (if signal is weak, node is far and hence should not be used as next hop candidate in routing). It can also assist data link layer protocols in link adaptive or hybrid Forward Error Correction (FEC)/Automatic Repeat Request
(ARQ) schemes [Tanenbaum and Wetherall]. Thus, a single source of information can be used to the improvement of many other protocols not directly related with the source of this information. Work in [Leung and Sung] suggest to increase transmission rate of node as required at the application layer when the channel quality is good at the physical layer. Protocols that use such interactions amongst layer; couple or merge logically separate layers are called cross layer protocols.

Srivastava and Motani [Srivastava and Motani] define cross layer design as, “protocol design by violation of a reference layered communication architecture is cross layer design with respect to the particular layered architecture”. This violation of layered architecture includes: creating new interfaces between layers; redefining the layer boundaries; designing protocol at a layer based on the details of how another layer is designed; joint tuning of parameters across layers; and so on. Cross layer protocols so designed impose some conditions on the processing at other layer(s).

A comprehensive definition encompassing all of existing cross layer approaches is given by Raja Jurdak [Jurdak, Wireless Ad Hoc and Sensor Networks: A Cross layer Design Perspective] as, “cross layer design with respect to a reference layered architecture is the design of algorithms, protocols, or architectures that exploit or provide a set of inter layer interactions that is a superset of the standard interfaces provided by the reference layered architecture”.

2.1.1 Cross Layer Approach

The cross layer design can be categorized into data sharing and design merger approaches as discussed next.

- **Data Sharing:** In this category of cross layer protocols adjacent or non-adjacent communication layers of the model (eg. Open System Interconnect (OSI) [Zimmermann]) share data through new unidirectional or bidirectional interfaces. Layering is preserved and enhanced with richer interaction. Support for richer interactions enables a closer coordination between the communication layers and optimizes the performance of protocol. Alternatively, a cross layer based model may support comprehensive state variables globally accessible to all layers. In both the cases the layered architecture is intact, which enables interoperability with layered architectures.
• **Design Merger:** In the extreme case, cross layer approaches can partially or completely merge functionality of protocols at different layers. Depending on the level of integration the merging of layers may result in monolithic design. Since layering is not maintained the architecture is not interoperable with layered ones.

### 2.1.2 Cross Layer Design Example

Many cross layer optimizations for traditional networks adopt data sharing approach which involves slight modifications to existing protocol stacks, and in turn yields significant improvements in performance.

Transmission Control Protocol/Internet Protocol (TCP/IP) (Socolofsky and Kale) stack has been proposed for wired connections, and hence in wireless environment it shows loss of performance. This is due to TCP’s failure in distinguishing network congestion from wireless errors, which leads to invocation of the congestion control mechanism for both cases (Gerla, Tang, and Bagrodia), (Mohammed et al.), (Fu et al.). One of the remedies to this problem TCP for wireless network, is expected to support an Explicit Congestion Notification (ECN) bit (Floyd). Routers at network layer can set the ECN bit to indicate that packets have experienced congestion. When the destination receives packets with ECN bit set to one, destination notifies sender of congestion so that the sender can reduce its transmission rate. Because the network layer at an intermediate node sets the ECN bit and transport layer at the destination reads it, the ECN bit represents a cross layer interaction.

Another example is the enhancement of the Mobile IP (Tanenbaum and Wetherall) hand off with link layer information. Hand off in Mobile IP networks occurs on detection of network changes at the network layer. Because network layer detection may be too slow, work in (Sanmateu et al.) propose the use of signal strength information from the physical layer on the active links to reduce the hand off latency.

### 2.1.3 Drivers for Cross Layer Design for Wireless Sensor Networks

This section formally presents the drivers that are motivations for cross layer design for sensor networks.
• Wireless Transmission Media: Wireless media possess some inherent adverse characteristics like: signal attenuation with increase in distance; multipath fading; high Bit Error Rates; hidden and exposed terminal problem; spatial contention and reuse; media capture effect; transmission interference and unfairness in media access. All these lead to packet losses which further require retransmission or error control and in turn increase energy requirements of the sensor node.

To address these problems simultaneously at all layers requires cross layer solutions (Vuran and Akyildiz). Transmission errors (at physical layer), QoS requirements (at application layer), retransmission (at transport/data link layer), error control (at transport/data link layer), and energy consumption (at physical layer) are closely related. When lower layers provide status of wireless link to higher layers, nodes can better adapt to its physical layer properties. For example, a routing protocol at network layer may be informed about the drop in quality of data received by data link layer so that it can create a new route to divert its traffic.

• Size, Resources, Energy: Depending on application requirements size of a single sensor node may vary from size of a brick to a dust particle. Varying size and cost constraint result in varying limits on energy, computation, storage, and communication resources. Nodes may have limited stored power source or may replenish it from environment (eg. solar cells).

Cross layer design approaches can expose power and computation related variables at several layers, enabling nodes to efficiently utilize their energy and computational resources. Protocols at each layer can see overall bigger picture of the layers in the node and make appropriate decisions.

• Application Specific Structure: The performance requirements of application vary greatly between different sensor network applications. For example, a sensor network for environmental monitoring prioritizes network lifetime to avoid power replenishment. In contrast, a sensor network for intrusion detection system emphasizes reliable and timely delivery. Cross layer can provide application specific performance requirements.
Furthermore, lifetime of WSN, the most common metrics for performance evaluation of WSN protocol is application specific. For applications where nodes are sparsely deployed, metric First Node Dies (FND) is used since network quality decreases considerably when one node dies. In a dense node deployment, metric Half of the Nodes Alive (HNA), which measures half of the life period of a WSN is used. Finally, in a redundant node deployment, metric Last Node Dies (LND), which gives an estimated value for overall lifetime of a WSN is used. Nevertheless, none of these metrics may be useful if the application tolerates certain amount of packet losses or a fixed delay \cite{Ozgovde and Ersoy}. Thus, the performance requirement of a WSN strongly depends on application.

- **Network Coverage and QoS:** Network Coverage is defined as ratio of monitored space to entire space.

  In case of dense and redundant deployment transmitting repeated data, results in increase in contention (MAC layer); congestion and complex routing (Network layer) thereby resulting in waste of energy (Physical layer). In a sparse deployment, more energy will be required to reach intermediate nodes and MS, resulting in network partitioning and decrease in network lifetime.

  To provide real time; secured; and private communication, protocols should offer services such as good coverage; congestion control; active buffer monitoring; acknowledgements; message cryptography; message priority; and packet loss recovery.

  All these require co-operation and information sharing among the layers which is possible through cross layering.

- **Time Synchronization:** Hardware clock reference of sensor node can directly affect an application operation. For instance, a WSN for target tracking is useless if it cannot register both position and detection time of an event. Thus, application layer depends on physical layer. The computation complexity of time synchronization protocol can directly affect network lifetime.

  This strong interdependence among layers of protocol stack in WSN is the major driver for using cross layer approaches for protocol design in WSN.
• **Mobility:** Sensor nodes may not change their initial location or may change it due to environmental effects like wind, rain etc. In case of tracking applications nodes may be carried by mobile entities. This may apply to only subset of nodes against all of them. Mobility causes parameter changes for the physical layer (e.g. interference signal levels), the data link layer (e.g. access schedules), the routing layer (e.g. topology change), and the transport layer (e.g. connection timeouts). With cross layering nodes can exploit interdependency between layers to manage their resources and hence the vital protocol parameters in mobile environments.

• **Distributed Network:** In contrast to a traditional infrastructure based network, where base stations have a global state information of the network, the network state in WSN is distributed across the nodes. Each node has a local view of network state that characterizes a partial view of the overall network state. In addition for most of the time, it is not possible to collect network state at a single node. This in turn prevents use of any centralized optimization algorithms. Thus, each node can use partial view of the network state to run distributed algorithms locally. Distributed algorithms can use a cross layer design to perform fine grained optimizations locally whenever it detects changes in state.

• **No Human Intervention:** Most sensor network applications work without human intervention. This requires the nodes to independently determine how to react to energy depletion, mobility, and loss of connectivity. Cross layer design provides nodes with details like, their existing resources contributing to autonomous and self configuration of the node.

• **Strong Interdependence among Layers:** When the nodes are densely deployed, number of transmissions in the network increases. This further increases the contention (at Medium Access Control (MAC) layer), congestion (at network layer) and results in waste of energy (at physical layer). In a sparse deployment, more transmission energy (at physical layer) will be required to reach intermediate relay nodes (at network layer) and to MS, resulting in network partitioning and decrease in network lifetime.
This strong interdependence among layers is the major motivation for using cross layer approaches for protocol design in WSN.

2.2 Cross Layer Protocols for Wireless Sensor Networks

A number of cross layer protocols and architectures have been proposed in the literature (Melodia, Vuran, and Pompili), (Mendes and Rodrigues), (Jagadeesan and Parthasarathy). Several representative protocols and architecture that are based on the cross layer design philosophy along with noteworthy findings are discussed next.

One of the most popular protocols, Low Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman, Chandrakasan, and Balakrishnan) combines energy efficient cluster based routing and media access together with application specific data aggregation. Clusters are formed by using a distributed algorithm and nodes make autonomous decisions whether to become a cluster head or not. Cluster heads are elected based on probability model taking care that a node becomes a cluster head in round \( r \), only if it has not been a cluster head for previous \((r - 1)\) rounds. Rotating cluster heads evenly distributes load among nodes. Use of TDMA for intra cluster communication makes node turn on its radio only when it has a transmitting turn, resulting in significant energy savings. However, this energy savings is sacrificed in cluster heads transmitting data directly to MS. Further, since each node probabilistically decides whether or not to become the cluster head, there might be cases when the selected cluster heads are in close vicinity of each other or from area with less node density like edge of network or may be having less residual energy sacrificing the overall performance of the network. As a result, after certain rounds there will be significant difference in energy consumption between the nodes near the MS and those far from the MS. If all nodes begin with the same initial energy, the nodes far from the MS will drain their energy faster than those near the MS resulting into network partition.

Self Organized TDMA Protocol (SOTP) combines routing and TDMA to serve the application specific and data centric needs of WSN (Wang et al.). More specifically, MS uses routing information for time slot allocation of node. To reduce data latency, nodes at distances far from MS are given a time slot at an earlier position in
the TDMA frame. The cross layer combination simplifies node design and achieves energy efficiency. To make network adaptive to node/link failures each node transmits an active state indicator packet at regular time intervals directly to MS. With this, nodes at long distance from MS end up using a large fraction of their energy reserves, which in turn decreases overall network lifetime. Above all, SOTP is shown to perform better compared to straightforward version of LEACH. This comparison is unfair, because SOTP uses multihop routing and in LEACH cluster heads communicate directly with MS.

Work in (Lu, Krishnamachari, and Raghavendra) presents Data gathering MAC (D-MAC), a cross layer protocol that combines media access with routing to constitute a data gathering tree rooted at MS. To enable uninterrupted data forwarding on the multihop path, D-MAC staggers schedule of nodes and allows them to wake up sequentially like a chain reaction. If $d$ is depth of node in data gathering tree, node sets its wake up schedule $d\mu$ ahead from MS schedule and periodically goes into receiving, sending and sleeping states. Thus, during no collision, data is sequentially forwarded to MS without sleep latency decreasing the data latency. Since nodes are awake only when it has to forward data, D-MAC decreases the energy consumption of nodes. The drawback of D-MAC is that it does not consider the node fairness and interference between nodes at the same depth is to be handled carefully. Further, protocol does not discuss how depth and route information will be known to the node. Finally, it is a route oblivious protocol and nodes outside routing paths waste energy in unnecessary idle listening.

Flexible schedule based TDMA Protocol (FlexiTP) uses knowledge of schedule of node at MAC protocol to enable routing protocol to disseminate and collect data to and from nodes (Lee, Datta, and Oliver). For network setup, it uses CSMA/CA (Kaveh and Levesque) and builds a data gathering tree rooted at MS. It assigns transmission schedules which are maintained by nodes throughout their lifetime. To avoid collision, a token passing scheme is used and nodes perform procedures only if they hold a token. After network setup, nodes perform data gathering tasks using their TDMA schedules. For this, nodes forward data to their parents until parents become unreachable due to energy depletion or adverse environmental conditions. Only the nodes which are involved in data transfer are awake during the TDMA slots.
which results into large energy savings. With loose slot structure, in which nodes can build, modify, or extend their schedules based on local information, FlexiTP guarantees end to end data delivery and fault tolerance. On the downside, number of child nodes a parent can have is not restricted, which results into uneven load distribution among parent nodes. Above all, the child selection is based on simple broadcast and reply mechanism which may lead to improper parent child pair and hence decrease overall reliability and throughput of the protocol.

Energy Efficient and Fast Forwarding (EEFF) is an asynchronous, energy efficient MAC protocol coupled with dynamic minimum hop routing selection (Tao et al.). The nodes are not synchronized and perform their active sleep schedule individually. The relay nodes decide whether to accept the request to send (RTS) from source nodes based on their local state (active or sleep) and data latency to MS. The candidate relay nodes then contend to send clear to send (CTS) to source nodes and the winner keeps awake to receive data. Nodes broadcast their local routing information periodically to build minimum hop routing table information for data transmission to MS through several hops. EEFF achieves low data latency and high energy savings, as its dynamic routing approach selects relay nodes during but not before the transmission and data is send without any preambles. On the downside, data latency estimation by relay node does not consider varying network traffic load, which might result into improper relay nodes. Above all, each data transmission requires RTS/CTS and acknowledgement (ACK) which increases overhead of the protocol (Xu and Saadawi).

Medium Access Control and Routing (MACRO) (Galluccio et al.) is an integrated MAC and routing protocol whose objective is delivery of data to a destination $D$, with known geographical coordinates while minimizing energy consumption. The cost for selecting the next hop is a weighted progress factor, which considers the progress toward destination per transmitted power. Suppose a sender node, $R'$, wants to contact destination node, $D$ and \{$P_1, P_2, \ldots, P_M$\} are the set of available power levels which can be used to transmit data packets with $P_1 < P_2 < \cdots < P_M$. Given the transmission power $P_j$, with $j \leq M$, $S_j(R')$ is the set of nodes in the radio coverage of the relay node $R'$, when the transmission power is $P_j$. With this, $S_1(R') \subseteq S_2(R') \subseteq \cdots \subseteq S_M(R')$. The selection of the next relay at each hop is
done such that the weighted progress factor $G_{R',R'}$, as defined in equation 2.1, is maximized.

$$G_{R',R'} = \frac{d(R', D) - d(R'', D)}{P}$$

where, $R'$ is the current relay node (or the source of information), $R''$ is the candidate next relay node, $P$ is the transmission power utilized by $R'$ for the delivery of the packet to $R''$, and $d(A,B)$ represents the distance between nodes $A$ and $B$. $G_{R',R'}$ represents the progress towards the destination $D$ obtained per unit of transmitted power. Once the next relay node $R''$ has been identified, the data packet is transmitted to $R''$. MACRO does not require location information to be exchanged in the network and each node is required to only know its own position and that of the destination reducing the energy consumption. To further decrease the energy consumption, the radio of the nodes is periodically turned off and different values of the transmission power levels are used. The criteria for selection of next hop increases the number of hops travelled by packets to reach the destination which might result in an increase of the end to end delay. Accordingly, MACRO can give best performance for applications where energy efficiency is a crucial requirement, and there are no strict requirements in terms of data latency. Finally, MACRO assumes the nodes in the network are poisson distributed which may not be the case in a real sensor network.

A Load Balancing Algorithm (ALBA), is a combined MAC and greedy geographical routing protocol for WSN that considers the congestion and traffic load balancing (Paolo et al., “ALBA: an adaptive load balanced algorithm for geographic forwarding in Wireless Sensor Networks”). To accomplish this each node which has data to send computes two values: Geographic Priority Index (GPI) and Queue Priority Index (QPI), indicating progress of node towards destination and its traffic load, respectively. Each node sends several RTS packets to scan QPI and GPI values of its neighbours. The neighbour whose values match the requested values responds to this packet by CTS packets. The sender node selects one of these neighbours as forwarder. Selecting routes with low congestion improves the performance of greedy routing much more than simply optimizing the geographical advancement. However, a packet that reaches a dead end gets stuck since the greedy rule adopted to select nodes closer to MS does not allow it to continue routing process through neighbours offering a negative advancement. This is a severe problem, particularly in topologies
that are connected but sparse.

A Load Balancing Algorithm-Rainbow (ALBA-R) (Paolo et al., "Efficient Non
Planar Routing around Dead Ends in Sparse Topologies Using Random Forwarding")
avoids dead end through a coloring scheme called Rainbow. On the basis of records of
its past success in finding forwarder, each node assigns itself a different color, which is
used to participate in communication. In the beginning, all nodes performing ALBA
basic operations are labeled "yellow". If node determines that it cannot forward
any packet after a number of attempts, it considers itself a dead end, turns to color
"red", and exponentially decreases its own likelihood to participate as an eligible
forwarder in contentions initiated by other (yellow) nodes. In this way, nodes located
on paths going to dead ends gradually realize that those paths are ineffectual for
routing, and stop volunteering as forwarders for other nodes. Selecting routes with
low congestion improves energy performance of protocol. However, the route selection
process incurs high overhead in calculation of parameters indicating progress of node
towards destination and its existing traffic load.

In (Laura Savidge), a distributed multihop routing scheme for application that
generates periodic packets is proposed. The cost function for the selection of next
hop is based on location, current queue lengths, and remaining energies of the neigh-
bouring nodes. The cost function is defined in equation (2.2) below,

\[ c(i) = c_p(i) + \alpha \times c_q(i) + \beta \times c_e(i) \]  

(2.2)

where \( c_p(i), c_q(i) \) and \( c_e(i) \) denote position cost, queuing cost, and remaining energy
cost, respectively. \( \alpha \) and \( \beta \) determine relative weights of the three node parameters.
The variable \( i \) denotes members in the set of neighbours for which the cost function
is evaluated. The neighbouring node with the lowest cost neighbour is chosen as the
next hop node. The weighted cost function allows flexibility in design tradeoffs like
that between average packet latency and network lifetime. On the downside, nodes
periodically broadcast a maintenance packet which includes its identification, current
queue sizes, and residual energy, which increases communication overhead.

Work in (Zamalloa et al.) selects the node with, maximum value of product of
packet reception rate (PRR) and advancement towards destination (DIST) for lo-
calized forwarding in lossy WSN with ARQ mechanism. It further introduces new
blacklisting and forwarding strategies based on distance; PRR; and PRR \( \times \) distance,
for geographic forwarding with and without ARQ scenarios. Simulation and experimental results show that relative blacklisting schemes (blacklisting of a node based on rank of node within a set of neighbours) reduces disconnections and achieves higher delivery rates than absolute blacklisting (blacklisting of a node done on the basis of specific threshold value) schemes, but this comes at the cost of lower energy efficiency. Further, the models used for analysis and simulation does not consider sources of energy consumption such as processing or sensing. All mathematical analysis for distribution of optimal forwarding distance is based on chain topology. Hence, a more general analysis with arbitrary topology network is necessary to confirm the optimality of the proposed scheme in WSN. Moreover, the work only focuses on applications of sensor networks in which: the flow is low rate; data reporting schedule is non overlapping or MAC collisions are at minimum (or nonexistent).

Cross Layer Routing Protocol (XLRP) is a cross layer protocol that presents an efficient routing strategy based on application layer information along with capabilities of physical layer (Gunasekaran and Qi). The application layer messages are classified according to their byte sizes as small and large and this information is made available to network layer to help it determine the next hop neighbour and accordingly switch transmitter power level. For large size messages, node switches to high transmission power levels and remains in lower power levels for small message transmissions. As the nodes switch to higher transmitter power levels, the transmission radius increases, increasing the number of receivers. Each receiver node sets up a signal strength threshold value and when it senses signals above the set threshold it considers itself an unintentional receiver and turns OFF its radio module saving its energy. Simulation results show that for increasing packet size, XLRP is energy and latency efficient compared to geographical routing. The energy saving is due to the reduction in the number of transmissions and controlled switching OFF of the receivers. Latency efficiency is due to decrease in number of hops along the transmission path. However, for small packet sizes (32 or 48 bytes), XLRP switches to low transmission power levels and hence energy and latency performance of XLRP is identical to geographical routing protocols.

Work in (De et al.) carries out a cross layer analysis of the effect of physical layer constraints on link layer and network layer performance in case of Code Division
Multiple Access (CDMA) based sensor networks. It discusses role of network topology in receiver design principles and the impact of network topology on end to end routing performance. Nevertheless, the work does not formally present a communication protocol for actual implementation of WSN. Further, CDMA technology may not be the most efficient scheme for the WSN because of its high computational requirement which increases the energy consumption of the nodes (Demirkol, Ersoy, and Alagoz).

Multihop Infrastructure Network Architecture (MINA) (Ding et al.) proposes clustered network architecture which groups nodes with same hop count to MS in the same cluster. It integrates TDMA based medium access with CDMA or Frequency Division Multiple Access (FDMA) for node communication. Transmission in MINA is organized as superframes with each superframe consisting of: (i) a MS broadcasting phase when MS broadcasts control packets and the data packets to the sensor nodes; (ii) network self organization phase during which nodes discover their neighbours, and their parameters and (iii) data transmission phase. For routing each node has a router at one layer near to MS. For routing, a simple forwarding mechanism is used, in which source node in a layer selects a node in the layer towards MS as its forwarding node to send the data packet. This node then forwards the packet to its forwarding node and so on until the packet reaches the MS. Thus routing protocol does not require excessive table or route maintenance overhead. The main drawback of MINA is that selection of a forwarding node is done only on the basis of remaining energy level, which results into a single node being selected as forwarder node by all its neighbouring nodes, severely depleting its energy.

Work in (Cui, Goldsmith, and Bahai) propose joint routing, MAC, and link layer optimization. They present a variable length TDMA scheme with M-ary Quadrature Amplitude Modulation (MQAM) modulation (Rappaport). Slot duration of each node is determined based on its traffic pattern and its distance to the neighbouring nodes in the network. These values are taken as inputs for calculating the optimum number of slots for each pair of nodes in the network and for providing relevant routing information. The protocol is best suited for applications with predefined traffic pattern.

Energy Aware Adaptive Low Power Listening (EA-ALPL) framework proposed in (Jurdak, “Modeling and Optimization of Ad Hoc and Sensor Networks”) combines
three aspects of node: (i) number of descendants in routing tree to consider traffic load of the node; (ii) radio duty cycle that determines how busy the node is (iii) role of node to consider its sensing and processing power consumption. EA-ALPL runs on each node with neighbourhood state information of node as input and adjusts MAC layer listening mode and routing cost of neighbours. Although the cross layer scheme is insightful, work does not propose a communication protocol for practical implementation but validates framework using B-MAC protocol (Polastre, Hill, and Culler).

MAC-CROSS protocol (Changsu, Ko, and Son), utilizes routing information at network layer for MAC layer to maximize sleep duration of each node. It uses network allocation vector (NAV) (Kaveh and Levesque) information in RTS/CTS packets to awaken only a subset of nodes along the routing path from source to MS. More specifically, when a node receives RTS packet including the final destination address of sink, its routing agent refers to the routing table for getting the next node towards sink and informs back to its own MAC. The MAC agent of the node transmits CTS packet including the next node information. After receiving the CTS packet, node along the path to sink changes its state as upcoming node towards sink and remains awake even when NAV timer expires for receiving data. Rest of the nodes sleep even if their NAV timer expires, to save their energy. However, use of RTS/CTS increases the overhead of protocol.

The Graded Backoff Flooding protocol proposed in (Zhenzhou and Hu) uses received signal strength at physical layer to decide backoff time for MAC layer and thereby reduce duplicate packet transmissions and collisions. However, flooding protocols are energy demanding (Tanenbaum and Wetherall) and hence used in limited WSN applications.

Distributed Activation with Predetermined Routes (DAPR) is a joint routing and node activation protocol with target application area as coverage (Perillo and Heinzelman). DAPR assigns an application cost that decides the importance of a node to the coverage task as: higher the cost, more important a node is to the coverage of a sparsely monitored area. Before routing of the packets, MS floods a query, which sets the routes with smallest cumulative costs. Packets are then routed along smallest cost paths. Nodes whose coverage are not required turn OFF, with
the ones having higher cost have the first chance to turn OFF. Thus with cross layer improvements, network can serve the application longer by exploiting nodes in densely deployed areas. However, the network model considered in the work is one in which sensors make a binary decision of whether to turn on or off depending on their quality of coverage. This kind of network model might not work in real world sensor nodes involving multiple sensors.

The Cross Layer data Dissemination protocol described in (Mudasser, Gondal, and Dooley) combines energy depreciation rate, node distance, and neighbourhood information from physical layer, together with TDMA schedules from MAC layer and network lifetime requirements from application layer, to determine most efficient routes to MS. Results confirm that the technique balances load on forwarding nodes and minimizes data latency. However, implementation of the protocol requires heavy recursions which demands high computation power and energy from the node.

In Convergent MAC (CMAC) (Liu, Fan, and Sinha), anycast is used to initially transmit packets to relay nodes. Candidate receivers carry out receiver based contention by prioritizing their CTS transmissions based on their routing metrics (which is latency to MS). Routes are selected based on the prioritization mechanism. CMAC converges from anycast to unicast once it establishes contact with a receiver having good routing metric.

In (Cui et al.), a joint routing, MAC, and link layer optimization framework is proposed. Although the optimization framework is intuitive, a protocol for real world implementation is not developed.

In (Lin, Shroff, and Srikant), an in depth analysis of optimization techniques for cross layer design in wireless networks is carried out. It concludes that bottleneck in optimization is scheduling due to the nonconvex nature of the scheduling problem. Subsequently, in (Lin and Shroff), a distributed cross layer congestion control and scheduling algorithm is proposed. However, the studies provide analytical results without any communication protocol design.

In (YuPin and Feng), an Adaptive Network Allocation Vector assisted Routing (ANAR) protocol is proposed to lighten the network congestion based on the information from the link layer. Each node uses the NAV information from the RTS and CTS packets generated by the MAC protocols (like IEEE 802.11 (Calhoun, Montemurro,
To calculate the probability of the channel being busy. The channel busy probability inherently considers: (i) the data rate/size of the transmitted packets along the interfering paths and (ii) the number of the interfering neighbouring nodes. The probability of channel being busy is then used during the route discovery process to determine efficient congestion free route. The protocol is best suited for MAC schemes utilizing NAV and RTS/CTS mechanisms and applications with high network congestion. However, the routing protocol presented is a variation of Ad hoc On Demand Distance Vector (AODV) protocol (Perkins and Royer) that requires large number of control packets to be sent during route discovery. Moreover, AODV is an address centric protocol and WSN opt for data centric routing protocols. Finally, MAC layer protocols involving RTS/CTS packets increase the control communication overhead (Xu and Saadawi).

Cross layer energy efficient routing proposed in (Chilamkurti et al.) uses signal strength information from IEEE 802.11 MAC layer (Calhoun, Montemurro, and Stanley) to distinguish between packet loss due to congestion or channel error and accordingly carry out route discovery in Dynamic Source Routing (DSR) protocol (Johnson, Hu, and Maltz). More specifically, after a certain number of attempts if a transmitter node is not able to communicate with its neighbour, MAC checks strength of last received signal from its neighbour. If signal strength is equal to or greater than the threshold, it is concluded that there is no link failure and DSR is informed not to initiate new route discovery. This reduces unnecessary route maintenance operations invoked by DSR protocol, resulting in significant energy savings. However, IEEE 802.11 is basically designed for single hop network and there are questions about its performance in multihop networks like WSN (Xu and Saadawi). Further, DSR is an address centric protocol and WSN opt for data centric routing protocols.

In (Kashif et al.), Self organized Ant Colony Optimization based routing algorithm that discovers optimal route with decision metrics based on velocity, packet reception rate, and remaining energy is used to decide the relay node. The simulation results show that for a grid topology the proposed algorithm shows better throughput and energy compared to Temporally Ordered Routing Algorithm (TORA) protocol (Vincent and Corson). However, in real world WSN implementation nodes are most of the time deployed in a random manner.
In (DeCleene et al.), Power Aware Random Scheduling (PARS) is proposed that integrates pseudo dynamic scheduling at the MAC layer with diversity routing at the network layer. The MAC layer protocol provides a complete control of the state of radio at physical layer. The MAC and routing based cross layering reduces significant overhead that occur due to RTS/CTS packets. However, it requires precise time synchronization among neighbouring nodes.

Work in (Li and Bartos), proposes Reactive Store and Forward (ReSaF) that implements a routing algorithm based on conditional forwarding to route packets on per block basis. A node broadcasts route request when it has packets to forward and waits for route replies from neighbours. Route replies consist of routing information of neighbours like: available energy; number of packets waiting to be forwarded; size of available buffer space; and probability of successful delivery of packets to MS. Sender picks the next hop based on two routing metrics: delivery probability and a routing score (ratio of available energy and current queue length) of a node. The conditional forwarding directs packets on a path with higher delivery probability, less congestion, and more available energy. ReSaF is shown to exhibit lower loss rate, takes less number of hops per packet delivery, and consumes less energy compared to collection tree protocol (CTP) (Omprakash et al.) and Hop (Ming et al.). This comparison is not fair, since both CTP and Hop are not cross layer protocols. Further, ReSaF is designed for mobile WSN and its performance in large size static networks with high communication link variability is left as future work by the authors.

Cross layer MAC protocol (CL-MAC) (Mohamed, Canli, and Khokhar) is designed to handle multipacket, multihop, and multilow traffic patterns while adapting to a wide range of traffic loads. Scheduling in CL-MAC is done using Flow Setup Packets (FSP) serving as an RTS packet to the destination node and as a CTS packet to the source node. FSP efficiently utilizes routing information like routing layer buffer and all flow setup requests from neighbours to transmit multiple data packets over multiple multihop flows. Due to this, CL-MAC makes more conversant scheduling decisions which considers the recent network status and based on it optimizes its scheduling mechanism. However, the protocol is best suited for heterogeneous WSN where: a single node incorporates several sensors; or different types of nodes (each monitoring a different phenomenon) are networked together.
Work in (Francesca, Abbagnale, and Cipollone) proposes Personal Area Network coordinator ELection (PANEL), a cross layer standard compliant procedure to self configure IEEE 802.15.4 (IEEE-Task-Group-4) based WSN. PANEL combines the network formation procedure defined at MAC layer by IEEE 802.15.4 standard with topology reconfiguration algorithm operating at network layer, to elect a personal area network (PAN) coordinator in a distributed way. Simulation results show that PANEL minimizes average number of hops between nodes of the network and PAN coordinator, thereby reducing the data latency and energy consumption compared to IEEE 802.15.4 standard. However, the proposal is limited to WSN using IEEE 802.15.4 standard.

As discussed in section 1.3.2 a WSN is distributed across nodes; there is no fixed infrastructure control and nodes are constrained in energy, storage and computational resources. Under these circumstances, a clustering architecture in which geographically adjacent nodes are organized into “clusters” with one node acting as cluster head and rest as cluster members is a prominent architecture. Clustering reduces collision among cluster members by coordinating their media access; balances load by rotating cluster head; reduces information updates (as node deaths and joins in a cluster need to be updated only by cluster members); offers scalability and spatial reuse (as non-neighbour clusters may use same frequency or code for transmission) (Heinzelman, Chandrakasan, and Balakrishnan). A variety of clustering protocols have been proposed in the literature (Jane and Chong), (Abbasi and Younis) (Deosarkar, Yadav, and Yadav) (Xuxun) (Sudhanshu and Kumar) (Gajjar, Sarkar, and Dasgupta, “Performance analysis of clustering protocols for Wireless Sensor Networks”). Some of the noteworthy clustering protocols are discussed in the subsequent paragraph.

Energy Efficient Clustering Scheme (EECS) for self organizing WSN uses probability function with following parameters for cluster head selection: (i) energy possession rate (current energy against initial energy); (ii) time for: cluster head selection, data aggregation, and transfer to MS; (iii) number of times the node is selected as cluster head (Kyubg et al.). These parameters select the best cluster head in terms of energy and communication cost of the node. However, cluster head is required to directly transmit the aggregated data to MS. This increases energy consumption of the cluster heads and decreases protocol scalability.
Several computational intelligence techniques that can adapt to complex and changing WSN environments have been proposed for cluster head selection and routing in WSN (Murtuza, Ang, and Seng), (Wenjing and Zhang). Some of these techniques are: fuzzy logic; neural networks; reinforcement learning; swarm intelligence; evolutionary algorithms; artificial immune systems and reinforce learning (Raghavendra, Forster, and Venayagamoorthy). From among these techniques, fuzzy logic is one of the best problem solving control system methodologies that provides a simple way to arrive at a definite conclusion with imprecise, non-numerical, noisy, or missing input information (Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes”), (Dubois and Prade), (Chou and Chung). For this, fuzzy logic uses heuristic human reasoning to deal with contradictory situations and imprecise data. Fuzzy sets used to draw conclusions and make decisions differ from classical sets, in that they do not have a crisp boundary which allows an object to be a partial member of a set. Fuzzy logic system describes the dynamic behavior of a system with a set of linguistic fuzzy rules derived from the knowledge of a human expert. This makes the fuzzy decision mimic how a person would make decisions, only much faster. The capability of fuzzy logic is exploited in the technical fields like: image, speech and signal processing; aerospace, robotics and embedded systems industries; along with non-technical fields like business, sales, and marketing (Zadeh, “Is there a need for fuzzy logic?”). The advantages of using fuzzy logic for WSN protocol design are:

- The terms used in WSN protocol’s performance (“lower latency”, “longer lifetime”) and transmission media characteristics (“less noisy”, “more busy”) have a fuzzy component (multi valued logic) making their fuzzy representation easy and realistic. Further, knowledge derived from experimental and simulation results in field of WSN can be incorporated in fuzzy rules.

- Mathematical modeling of the system is not required for fuzzy control system design. Mathematical modeling is very difficult, if not impossible for WSN protocols as it involves multiple input variables with each of them defined separately for different WSN scenarios.

- Being a prescribed methodology to imitate the intelligent decision making pro-
cess of a human expert, fuzzy logic control can easily and efficiently deal with various uncertainties of WSN such as unreliable media and unpredictable changes in network topology to arrive at a definite conclusion.

• Fuzzy logic is flexible, scalable, fault tolerant, requires less system development cost, resources (computation and memory requirements), and design time (Kim, Ahn, and Kwon) compared to other computational intelligence techniques.

Due to the above mentioned advantages, fuzzy logic is used in WSN for: clustering and cluster head selection (Gajjar, Sarkar, and Dasgupta, “Cluster Head Selection Protocol using Fuzzy Logic for Wireless Sensor Networks”); location updating (Chou and Chung); routing (Kim and Cho), (Chiang and Wang), (Perianu and Havinga) (Minhas, Gopalakrishnan, and Leung); improving accuracy of event detection (Liang and Wang); security (Lee and Cho); and QoS support (Xia et al.). Several representative clustering and cluster head selection protocols that use fuzzy logic are discussed next.

Clustering protocol by Indrail et. al. uses fuzzy logic with input variables as: energy of node; concentration (number of neighbouring nodes) and centrality with respect to cluster for cluster head selection (Gupta, Riordan, and Sampalli). Cluster head selection protocol by Ran et. al. uses fuzzy logic with: energy of node; its distance from MS; and network density, for cluster head selection (Ran, Zhang, and Gong). In both the above mentioned protocols, fuzzy logic selects best cluster head in terms of energy of node and communication cost. However, since the cluster head selection algorithm is run at MS, each node is required to send its current location and energy level to MS in each round. In a real world sensor network, this may not be possible due to energy limitations of the node. Even if it is possible, all the nodes may end up using a significant fraction of their energy reserves in transmitting their parameters to the MS.

Cluster Head Election mechanism using Fuzzy logic (CHEF) (Kim et al.) protocol overcomes overhead of node sending its parametric information to MS by running fuzzy logic locally at the node in a distributed manner. Nodes in CHEF generate a random number and if this number is less than a predefined threshold, the node becomes a tentative cluster head. Final cluster heads are selected from tentative cluster
heads by running fuzzy logic with: energy of node and its local distance (sum of dis-
tances between particular node and its neighbours within a specified radius) as fuzzy
descriptors. Since tentative cluster heads in CHEF are selected randomly, there are
chances that some capable nodes might lose their chance to become a cluster head.
Further, cluster heads are not well distributed in the field. Consequently, some nodes
find themselves uncovered (orphan nodes), and are required to send their sensed data
directly to the MS which increases their energy consumption.

Work in [Leonard et al.] proposes intelligent fuzzy based cluster head selection
protocol called F3N that uses fuzzy logic with: residual energy; network traffic; num-
ber of neighbouring nodes; and distance from cluster centroid as fuzzy descriptors for
cluster head selection.

Fuzzy Relevance based Cluster head selection Algorithm (FRCA) proposed in
[Lee and Jeong] uses fuzzy logic with: residual energy; signal strength of cluster
members; and distance between cluster head and cluster members as fuzzy descriptors
for cluster head selection.

Work in [Rogaia et al.], presents Fuzzy Logic Cluster Formation Protocol (FLC-
FP), which uses fuzzy logic with: energy of cluster head; distance between MS and
cluster head; and distance between cluster head and nodes as fuzzy descriptors for
cluster formation. FLCFP differs from the above mentioned protocols in that the
non cluster heads run fuzzy logic and join cluster head with maximum chance value
(fuzzy output) to form the cluster.

In [Hoda et al.], an energy aware distributed dynamic clustering protocol (ECPF)
is proposed. Each node in ECPF waits for a time interval inversely proportional to
its residual energy before competing for becoming a cluster head or joining a clus-
ter. Thus, nodes with high residual energy will become tentative cluster heads. The
final cluster head selection is done using fuzzy logic with node degree (neighbouring
nodes divided by total number of nodes) and node centrality in a given area. Finally,
clustering is performed only when energy of cluster head goes below a threshold value.

In [Nasim, Akbarzadeh, and Yaghmaee], a two level fuzzy logic is used to de-
termines a cluster head. In the first level, called local level, cluster heads are selected
using fuzzy logic with energy and number of neighbouring nodes as fuzzy descriptors.
In the second level, called global level, centerness of node with respect to its cluster;
distance from MS; and distance from other cluster heads are used as fuzzy descriptors for selecting cluster heads from amongst those selected in local level.

Energy Efficient Dynamic Scenario (EEDS) \cite{DuttaNaskarMishra} proposes cluster head election using fuzzy logic with: energy and number of neighbours as fuzzy descriptors for cluster head selection at local level. It then uses fuzzy logic with: transmission and residual energy; energy consumption rate; queue size; centrality and proximity to the MS, as fuzzy descriptors for cluster head selection at global level.

Fuzzy logic for cluster head selection in F3N \cite{Leonard}, FRCA \cite{LeeJeong}, FLCFP \cite{Rogaia}, ECPF \cite{Hoda}, two level fuzzy logic based cluster head selection \cite{NasimAkbarzadehYaghmaee} and EEDS \cite{DuttaNaskarMishra} selects the best cluster head in terms of energy and intra cluster communication cost. However, these protocols do not concentrate on optimizing cluster formation. Further, the selected cluster heads in all the protocols are required to transmit cluster aggregated data directly to MS in each round. Thus, the cluster heads end up using a large fraction of their energy reserves in transmitting data directly to MS.

As a solution to this problem, one might think of using inter cluster multi hop routing techniques available in the literature instead of direct data delivery to MS by the cluster heads. This will decrease overall network energy consumption and increase scalability of the protocol compared to single hop approach.

One of the initial methods of inter cluster multihop routing is Minimum Transmission Energy (MTE) routing \cite{Ettus}, where intermediate routing nodes are chosen such that sum of squared distances (and hence total transmit energy, assuming a $d^2$ power loss \cite{Rappaport}) is minimized. With MTE, nodes near MS work as relays with higher probability compared to far nodes making them die earlier and resulting into hot spots near MS.

Energy Efficient Hierarchical Clustering (EEHC) is a distributed randomized clustering algorithm that divides WSN into clusters with a multilevel hierarchy of cluster heads rooted at MS \cite{BandyopadhyayCoyle}. At each level, node becomes a volunteer cluster head with probability $p$ and advertises itself as a cluster head to nodes within $k$ hops communication range. Authors compute optimal values of $p$
and $k$ parameters that guarantees minimum network energy consumption. On the downside, multi level hierarchy of clusters results into more relay traffic for nodes near the MS \cite{Mhatre and Rosenberg}. Further, the work does not consider an underlying routing and MAC protocol to see how they affect values of optimal probabilities of becoming a cluster head and run time of the algorithm.

Hybrid, Energy Efficient and Distributed (HEED) protocol proposed in \cite{Younis and Fahmy} is an iterative clustering algorithm. Both election of cluster heads and joining of clusters is performed based on the hybrid combination of two parameters. The primary parameter depends on residual energy of node and the alternative parameter is intra cluster communication cost. If power level for intra cluster communication is fixed for all the nodes, communication cost can be proportional to (i) node degree, when application requires load distribution between cluster heads, or (ii) $\frac{1}{\text{node degree}}$, when it is required to produce dense clusters. If variable power level is permissible, AMRP, the average of minimum power levels required by nodes within cluster range to access the cluster head is used as communication cost. In this approach, every regular node elects the least communication cost cluster head to join it. The cluster heads send aggregated data to MS in a multihop fashion. HEED is a fully distributed clustering approach that is benefited by use of two parameters for cluster head election. Further, the probability of two nodes within each other’s transmission range becoming cluster heads is negligible. HEED suffers from large transmission overhead, since its clustering process requires several iterations and a lot of control packets are broadcast to update neighbour sets in each iteration.

Rotated HEED (RHEED), a modified version of HEED is proposed in \cite{Wail et al.}. The first round of the setup phase of RHEED is similar to HEED. For the upcoming rounds, the clusters are fixed and only cluster head nodes are rotated among cluster members based on 'cluster head turn schedule' prepared in the first round. This avoids re-election of cluster head in all the rounds and saves network energy compared to HEED. During the cluster head rotation if the residual energy of cluster head goes below a specific threshold the network is re-clustered. However, the rotation of the cluster heads in RHEED is completely random. This might lead to improper cluster head selection which in turn decreases network lifetime.

Distributed Weight based Energy efficient Hierarchical Clustering (DWEHC) is
another distributed clustering protocol that constructs multilevel clusters with one
cluster head and its first level child node, second level child node, and so on \cite{Ding, Holliday, Celik}. The number of levels within one cluster is determined by clus-
ter radius. TDMA is used for intra cluster communication and 802.11 based MAC
mechanism \cite{Tanenbaum and Wetherall} is used by cluster heads for data transmis-
sion to MS. The nodes have a limited number of child nodes, which makes DWEHC
scalable and cluster heads are distributed such that probability of two nodes within
each other’s cluster range becoming cluster heads is very less. The main drawback
of DWHEC is that cluster heads use 802.11 based MAC mechanisms which is princi-
 rally designed for single hop network and there are questions about its performance
in multihop networks like WSN \cite{Xu and Saadawi}.

Position based Aggregator Node Election (PAGNEL) is a position based cluster-
ing routing protocol for WSN \cite{Buttyn and Schaffer}. It assumes that the nodes are
deployed in a bounded area which is separated into geographical clusters. Clustering
is pre-determined before the deployment of the network, and each node is pre-loaded
with the geographical information of its cluster. At the start of each epoch, nodes
in each cluster calculate a reference point depending on the epoch number in a dis-
tributed manner. Then after, the nodes in the cluster elect the node that is closest to
the reference point as cluster head for the given epoch. In each epoch, the reference
points of the clusters are re-calculated and the cluster head election is re-executed.
This is done for load balancing, as each cluster node can become cluster head with
nearly equal probability. The communication overhead used in the cluster head elec-
tion is also used to maintain the routing tables within the cluster. At the end of
cluster head election phase, the nodes learn about their next hop neighbours towards
their cluster head elected for the current epoch. For inter cluster communication, the
authors suggest to integrate PANEL with any position based routing protocol like the
Greedy Perimeter Stateless Routing (GPSR) protocol \cite{Karp and Kung}. Data aggre-
gation in PANEL significantly reduces the number of transmissions and receptions.
However, PANEL assumes that clustering is pre-determined before the deployment
of the network and hence cannot adapt to the dynamics of WSN.

Power Efficient GAthering in Sensor Information Systems (PEGASIS) uses a
chain structure to transmit data to MS \cite{Lindse and Raghavendra}. Nodes in PEGA-
SIS are assumed to: have global knowledge about location of other nodes; can control their transmission power; and are equipped with CDMA-capable radio transceivers (Lindse, Raghavendra, and Sivalingam). The construction of the chain starts with farthest node from MS called end node. Nodes are added to the chain gradually using a greedy method, starting from the closest neighbour to the end node. This is done first for the closest neighbour to the top node in the current chain and continued until all nodes are included. Signal strength is used to measure distance to neighbours and hence determine the closest neighbour. A node within the chain is selected as chain leader and given the responsibility to transmit aggregated data to MS. The role of leader is changed regularly by the MS to balance energy consumption of the network. PEGASIS decreases the overall energy consumption of the network by the following strategies: (i) use of multihop routing chain to decrease the transmission distance of a non cluster head (ii) elimination of dynamic cluster formation overhead (iii) selection of only one node instead of multiple nodes for data transmission to the MS and (iv) decrease in number of transmissions and reception by data aggregation. However, nodes assuming role of chain leader may be arbitrarily far from MS. Such a chain leader may be required to transmit with high power to reach the MS. Above all, a single chain leader can be a bottleneck. Finally, PEGASIS introduces excessive delay in data transmission for node at long distant from MS in the chain.

Hierarchical PEGASIS (Lindse, Raghavendra, and Sivalingam) is an expansion of PEGASIS that aims at decreasing the delay incurred for packet transmission to the MS. For this, simultaneous transmissions of data packets are done and energy × delay metric is considered for data gathering in the network. Two approaches are suggested to avoid collisions and interference among nodes. The first approach is to use signal coding like CDMA (Rappaport) and second approach is to allow simultaneous transmissions of spatially separated nodes. In the first approach, chain of nodes forms a tree like hierarchy with each selected node in a particular level transmitting data packets to the node in the upper level of the hierarchy. This method ensures parallel data transmission and significantly reduces the delay to \(O(\log N)\) where \(N\) is the number of nodes. In the second approach, a three level hierarchy of the nodes is created and node interference is reduced by properly scheduling simultaneous transmissions. Overall the Hierarchical PEGASIS protocol is shown to perform superior than orig-
inal PEGASIS by a factor of about 60. However, every node needs to be aware of its neighbourhood to route the data packets, which incurs significant transmission overhead especially when the network is highly utilized.

Work in (Sungju et al.) proposes Energy Efficient Distributed Unequal Clustering (EEDUC) protocol for WSN. Nodes in EEDUC compete to become cluster head and the one with large number of neighbourhood nodes becomes the cluster head. The cluster heads then calculate the cluster head competition radius which is a function of: energy of node; its distance from MS and number of neighbouring nodes. Thus, competition will form unequal sized clusters and balance the energy consumption of the network.

In (Jiguo et al., “A cluster based routing protocol for Wireless Sensor Networks with nonuniform node distribution”), a protocol that includes an Energy Aware Clustering algorithm (EADC) and a cluster based routing algorithm is proposed. The cluster heads in EADC broadcast cluster head advertisement using the same competition range to construct clusters of even sizes. This balances the energy consumption among cluster members but imbalances the energy consumption among cluster heads due to non uniform distribution of nodes. Thus, cluster heads in dense areas may have more cluster members, increasing their intra cluster energy consumption. To overcome this, the inter cluster multihop routing protocol selects relay cluster head with higher residual energy and a smaller number of cluster members to balance the energy consumption among cluster heads.

Use of multihop routing for inter cluster communication in EEHC (Bandyopadhyay and Coyle), HEED (Younis and Fahmy), RHEED (Wail et al.), DWEHC (Ding, Holliday, and Celik), EEDUC (Sungju et al.) and EADC (Jiguo et al., “A cluster based routing protocol for Wireless Sensor Networks with nonuniform node distribution”) leads to hot spots problem. To mitigate this problem, protocols with unequal cluster size to evenly distribute traffic load amongst cluster heads are proposed in the literature (Selvi and Manoharan).

Work in (Enver et al.) proposes an unequal clustering protocol based on HEED called Unequal HEED (UHEED). Each cluster head in HEED has similar cluster head competition radius which leads to uniform clusters throughout the network. The cluster head competition radius in UHEED is a function of distance from MS that
ensures smaller size clusters near MS. This in turn, mitigates the hot spots problem; uniformly distributes energy across the networks; and increases the network lifetime. However, UHEED being an extension of HEED suffers from all the disadvantages of HEED.

In (Guihai et al.), Unequal Cluster based Routing (UCR) protocol that groups the nodes into clusters of unequal sizes is presented. The tentative cluster heads in UCR are selected randomly. These nodes then compete for becoming final cluster heads. The cluster head competition radius of each tentative cluster head is different and is the function of its distance from MS. Final cluster head within a competition range is selected based on the residual energy. For inter cluster routing the relay node is selected based on residual energy of the node.

In (Yu, Qi, and Wang), an Energy Driven Unequal Clustering protocol (EDUC) for heterogeneous WSN is proposed. The cluster head competition radius is a function of distance of cluster head from MS. The function ensures that the radius is less for nodes near MS and more for nodes far from MS. However, the initial set of cluster heads are selected only on the basis of their residual energy, which is not a sufficient parameter for cluster head selection.

Improved Fuzzy Unequal Clustering (IFUC) protocol uses energy, distance to MS, and local density of the node as fuzzy descriptors to: select cluster heads; and also estimate cluster head competence radius (Song et al.). The tentative cluster heads are selected randomly, which then run fuzzy logic for final cluster head selection. Ant Colony Optimization (Marco and Stutzle) algorithm is used to find the optimal paths from cluster heads to MS by considering the inter cluster traffic. The protocol balances energy consumption across the network and increases the network lifetime.

Work in (Bagci and Yazici) presents Energy Aware Unequal Clustering using Fuzzy logic (EAUCF), a fuzzy based distributed clustering protocol. EAUCF uses a probabilistic model for tentative cluster head election. Each tentative cluster head then calculates cluster head competition radius using fuzzy logic with residual energy and distance from MS as fuzzy descriptors. The cluster head selection algorithm takes care that, nodes with high residual energy and those far from MS have a very large cluster head competition radius; and those with less residual energy and near the MS have very small competition radius. Unequal clustering distributes the traffic load
among all sensor nodes evenly and increases the network lifetime.

Work in (Li et al., "COCA: Constructing Optimal Clustering Architecture to Maximize Sensor Network Lifetime") proposes Constructing Optimal Clustering Architecture (COCA), a unit based clustering architecture to maximize network lifetime. The work presents theoretical analysis for calculating optimal number of units in the network field that minimizes overall energy consumption of the network. It then proposes an iterative algorithm that determines number of clusters within each unit. The tentative set of cluster heads are selected randomly by MS and the final cluster heads are selected based on residual energy. The inter cluster routing protocol selects the relay node based on residual energy.

Tentative cluster head selection in EAUCF (Bagci and Yazici) is done using a probabilistic model; and in UCR (Guihai et al.), EDUC (Yu, Qi, and Wang), IFUC (Song et al.) and COCA (Li et al., "COCA: Constructing Optimal Clustering Architecture to Maximize Sensor Network Lifetime") it is done randomly. Thus, there are chances that nodes with unfavorable parameters like less residual energy, far from MS, or having less neighbouring nodes might become final cluster heads. This in turn deteriorates the network lifetime of these protocols.

In Unequal Layered Clustering Approach (ULCA) (Zhao and Wang) the network is divided into layers such that the layers close to the MS are of smaller size compared to layers far from MS. The MS then generates layer specific parameters like number of nodes, expected number of cluster heads, radius of cluster, and initial probability of becoming cluster heads. These parameters are then broadcast to the entire network. Nodes in the same layer then compete to become cluster heads and the one with maximum residual energy becomes the cluster head. The inter cluster routing protocol considers the residual energy and distance to MS as parameters for selecting the relay cluster head.

Work in (Wei et al.) proposes Energy efficient Clustering (EC), a distributed clustering algorithm that determines cluster sizes depending on number of hops to the MS. The probability of a node becoming a cluster head is tuned so that there is approximately uniform energy distribution across the network. However, the inter cluster routing protocol chooses the next hop cluster head from the fixed neighbour rectangular region and thus considers only the distance criteria for relay cluster head
In Energy Aware Distributed Unequal Clustering protocol (EADUC), nodes compete to become cluster head and the one with high residual energy becomes the cluster head (Jigu et al., "A cluster based routing protocol for Wireless Sensor Networks with nonuniform node distribution"). The cluster heads then calculate the cluster head completion radius which is a function of energy of the node and its distance from MS. The competition radius of cluster heads far from MS and with more residual energy is more compared to those near the MS and having less residual energy. For inter cluster routing the relay cluster head is selected based on residual energy of cluster head. The protocol is for heterogeneous networks and hence requires certain nodes with more energy (super nodes) positioned at some predetermined locations to control cluster sizes. However, in a real life WSN application this requirement would be seldom met with.

Work in (Babar and Hasbullah) proposes Energy Efficient and QoS aware Routing (EEQR) protocol for clustered WSN. To increase the network life time and minimize data latency, EEQR uses a combination of mobile and static MS. To mitigate the hot spots problem with static MS, delay sensitive packets are forwarded to the static MS, and delay tolerant packets are sent to mobile MS. Generally, mobile MS face the problem of increased latency as the nodes have to wait for the mobile MS to come in the vicinity of the data forwarding nodes. To address this problem, traffic is prioritized based on its QoS requirements and movement of mobile is associated with traffic priority. The protocol requires static as well as mobile MS for its operation. However, in real life WSN applications this requirement would be seldom met with.

Most of the WSN protocols discussed above focus on framework design and execution related matters. Historically, little attention is given to cross layer architectures which can standardize development and assist protocols in their functioning. Some of the remarkable cross layer architectures for WSN are discussed next.

### 2.3 Cross Layer Architectures for Wireless Sensor Networks

One of the initial steps towards design of WSN architecture is Sensor network Protocol (SenP) that nurtures greater interoperability between different code bases (Polastre
et al.) Work in (Cheng et al.) proposes a modular network layer for sensor networks that sits on SenP. Modular design approach increases protocol code reuse and allows several protocols to co-exist by run time sharing of code, memory, and radio. However, SenP does not address important issues like power management and reliability in WSN. Further, the interface between network and data link layer is quite complex hindering fast deployment of WSN.

A Cross layer Protocol for Efficient Communication in WSN (XLP) (Mehmet and Akyildiz) merges all layers from physical to transport. A node in XLP determines to take part in communication based on status of communication link, buffer occupancy, and energy. This ensures transmission reliability, manages network congestion, and assures uniform energy consumption across the network. Merging of layers guarantees coordination of their actions by benefitting from same up to date information. However, it is difficult to maintain and evolve the architecture without layers or standardized interface between them. Further, XLP works only for CSMA/CA (Kaveh and Levesque) based protocols and uses geographical routing that requires nodes to know their precise location. Finally, each data transmission requires RTS/CTS and ACK which increases overhead of the protocol (Xu and Saadawi).

X-Lisa (Christophe) is another cross layer information sharing architecture that maintains a repository of information required for operation of protocols at different layers. This is done by cross layer optimization interface which maintains a neighbour table (containing details of neighbours) and a message pool (details of messages send and received). With the availability of important network information at all layers in the stack, protocols may tune their internal parameters to improve network performance. However, maintenance and updating of the neighbourhood state involves a large communication overhead in X-Lisa.

TinyCubus (Lachenmann) describes cross layer framework that provides interfaces for information sharing among layers and enables lower layers to invoke application specific code. For information sharing, a state repository with all relevant cross layer parameters is defined. The invocation of application specific code is done by extending the operating system, TinyOS (Levis). Through the cross layer state repository, application and communication blocks work together to provide system specific optimization. Facility for running application specific code makes it a generic recon-
figurable framework but the architecture works only with component based operating systems like TinyOS.

Embedded Wireless Interconnect (EWI) (Liang and Hatzinakos) coordinates the nodes cooperation at application and physical layers. The physical layer supplies a library of wireless transmission modules to application layer which decides organization of wireless links by exploiting tradeoff between application specific QoS gain and energy consumption. Compared to OSI model (Zimmermann), EWI better exploits node redundancy and broadcasting nature of wireless media to minimize energy consumption of the network. However, the platform requires energy demanding routing protocols like flooding or gossiping (Yuecheng and Cheng).

XIAN (Herve et al.) provides MAC layer application programming interfaces (API) to arrange for bi-directional interaction between MAC and: network, transport, and application layers. 802.11 MAC (Calhoun, Montemurro, and Stanley) and physical layer metrics are measured through XIAN API and are used to improve quality of routes selected by routing protocols. However, the interactions between network and transport layers; or between transport and application layers, are impossible which limits capacity of XIAN. Further, the authors have proposed interfaces only for Linux operating system (Petersen), which makes XIAN operating system dependent.

Work in (Hyeok, Perry, and Nettles) presents architecture with components like: intra node connectors inside a single node (to integrate existing protocol modules); inter node connectors (for information exchange among nodes); and a cross layer processor that executes its synchronous or asynchronous adaptations outside protocol module. Separate modules for different functions provide: modularity, flexibility, extensibility, and portability.

XLEngine (Jomaa et al.) augments layered architecture with a cross layer manager component, which consist of cooperating components to: maintain and update node parameters; allow cross layer data exchange and selectively broadcast it. With local and network wide knowledge, XLEngine supports QoS in wireless networks. However, both the above mentioned architectures are tested on wireless networks with high power mobile devices and hence there has to be a check whether their computing overhead is suited for low power sensor nodes or not.

Optimizing Agent (OA) (Su and Lim) utilizes an OSI (Zimmermann) like stack
with an agent to exchange and control information between different layers. Each layer in the stack provides new interfaces to offload cross layer optimization to OA. Layer interactions are categorized as intra layer or inter layer interactions and can be either bottom up or top down. The architecture is simple and flexible. However, giving OA an unrestricted access to every layer can be unsafe and hence the new interfaces must be carefully defined and created. Further, the authors do not discuss the performance related inputs to the OA and its implementation details.

Cubic and Cross layer (CCL) (Chuan et al.) for sensor network enables cross layering with notification services and communication abstraction for data centric communication. Sensor service protocol (SSP) is used as a middleware layer that provides services for applications and keeps interface platform independent. Thus, CCL factors out functionality requirement of the application and composes them in a coherent structure, while allowing innovative technologies and applications to evolve independently. However, the work does not clarify the exact services and functionalities provided by SSP, and interaction between SSP and the layers.

Work in (Madan et al.) proposes a joint routing, MAC and link layer optimization framework. Authors begin by developing a model framework with sets of potential optimization objectives and system constraints. First, each of the three layers is individually optimized for three different objectives while the other two are held static. Next, all three layers are jointly optimized for a TDMA network, for cases where interference is and is not allowed. Computational results demonstrate the gains achieved by cross layer design. However, power consumption model for an active link in a TDMA slot does not include receiver circuit power consumption, which is not a realistic assumption.

Aqua Net (Zheng et al., “AquaNet: An Underwater Sensor Network Architecture: Design, Implementation, and Initial Testing”) is a generic architecture for underwater sensor networks that provides Berkeley Software Distribution (BSD) socket (Jorgensen) like interface that makes it easy for users to implement applications. The architecture allows cross layer design by providing system parameters across all layers but its usage is limited to underwater sensor network.

An Adaptive Cross Layer Sensor Network Architecture that enables multi-scale collaboration and communication with an Information Exchange Service (IES) is
presented in [Santashil]. IES acts as an information database and notification service that allows cross layer optimization by making information of one service available to another and by making application requirements available to the layers. However, the architecture does not provide information to be maintained by IES restricting the usage of “plug and play ”features. Further, by introduction of data fusion and data service layers the architecture loses its interoperability with OSI [Zimmermann] based architectures.

Cross LAyer Management Plane (CLAMP) [Ahmad] comprises of a database, services, and a set of interfaces. The database is implemented as a list of performance and energy aware network parameters. The layer that owns the parameter updates its value in the database. Service components notifies these updates to the interested layers which use them for their functions.

Horizontal Framework (HF) [Hakala and Tikkakoski] uses a horizontal architecture with Cross Layer Management (CLM) and Application and Protocol Stack (APS) components. CLM offers a shared data structure and takes care of sensor network specific functions like: topology control, routing, physical layer coding, and power saving. APS is responsible for application specific data transmission and services. HF simplifies the protocol stack by separating certain tasks as modules of CLM entity and reduced header information which makes application programming easier. However, from implementation view point, interoperability between CLM modules is not clarified in the work.

Work in [Mario et al] proposes an adaptive and cross layer framework for reliable and energy efficient data collection in WSN based on ZigBee [ZigBee-Alliance] standards. A low complexity distributed heuristic algorithm, called ADaptive Access Parameters Tuning (ADAPT) is developed which estimates current traffic conditions and tunes MAC parameters according to required application specific reliability (delivery ratio). Simulation results shows that ADAPT meets reliability requirements with low energy consumption. However, ADAPT uses a heuristic approach instead of computational intelligence techniques in its decision making process.

Score [AlOmari, Du, and Shi] framework provides three basic mechanisms and interfaces to facilitate network components collaboration. First, a unified neighbour set abstraction under which a network component can read or write any neighbour
record by pointing at the record and performing a read or a write using score access
interface. Second, a modular cross layer interface that decouples network components
by providing a mechanism for them to communicate without pair wise interfaces.
Third, Score supports cross layer coordination by maintaining current operational
state of a sensor node (e.g., discovery, booted, sleep, and active). Protocol at each
layer reacts in its own way based on the state of the node announced by Score.

Based on the functional proximity between layers and the frequency of interac-
tions between them, Triangular CL Architecture (TCLA) [Daidi and Li] merges OSI
(Zimmermann) model into three virtual functional modules namely: AppM (appli-
cation, presentation and session) which is responsible for application aspects; NetM
(transport and network) which manages resource allocation, routing and congestion
control; and LinkM (data and physical) that deals with signals, coding, multiuser
access, error control, power control, etc. These three virtual functional modules are
organized in a triangular way with layering retained within the functional modules.
Internal protocol layers of one functional module interact directly, while layers in
different functional modules interact through a communication agent in each virtual
functional module. Each agent consists of standard interface for transmission of stan-
dard packet for non cross layer protocols and shared interface to exchange shared
information between two layers anywhere in different function modules. TCLA com-
bines the layered stack and non-stack approaches so as to maintain compatibility
with existing layered architectures and also obtain performance gain by cross lay-
ering. However, the work does not suggest: (i) types of data structure that are to
be maintained by communication agent (ii) optimization algorithm residing in the
communication agent.

Lu’s framework [Lu and Krishnamachari] considers horizontal layer that encom-
passes traditional layer functionalities, and vertical layers that represent cross layer
interactions. It introduces: (i) horizontal connectivity maintenance layer between
network and MAC layers to maintain a connected network topology (ii) data man-
agement layer between the application and transport layers for data placement, data
discovery, and in-network processing (iii) coverage maintenance layer to guarantee a
sufficient number of nodes to monitor a target area (iv) location and time service
layer for node localization and time synchronization. The model maps application


QoS requirements to parameters at all other layers in the architecture. The drawback of this approach is its implementation complexity due to both horizontal and vertical interaction.

2.4 Publication Related to the Chapter


