We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,900
Open access books available

145,000
International authors and editors

180M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Literature Review of MAC, Routing and Cross Layer Design Protocols for WSN

Tayseer AL-Khdour, Uthman Baroudi
King Fahd University of Petroleum and Minerals
Saudi Arabi

1. Introduction

A WSN is composed of a large number of sensor nodes that are communicating using a wireless medium. The sensor nodes are deployed in the environment to be monitored in ad hoc structure. In WSN, there is sink node that collects data from all sensors, and usually not all nodes hear all other nodes. WSN is considered a multi-hop network.

Although a WSN is a wireless multi-hop network, the ease of deployment of sensor nodes, the system lifetime, the data latency, and the quality of the network distinguish WSN from traditional multi-hop wireless networks. These features must be taken into account when designing different protocols that control the operation of WSN such as MAC protocols and routing protocols. Therefore, Many MAC and Routing protocols are proposed for WSN. These protocols take into account the distinguished features of WSN. Moreover, Cross layer design protocols are proposed for WSN. In cross layer design protocols, different layers interact to optimize the performance of the WSN protocol.

In this chapter, we will present a survey of the most well known protocols for WSN. A survey of the most well-known MAC protocols is presented in section 0. Section 0 presents discussion of routing protocols of WSN and classification of these protocols according to data traffic models. The routing protocols are also classified as: data centric protocols, hierarchical protocols, location-based protocols and QoS-aware protocols. In section 0, we will present some cross layer design protocols for WSN. A summery of the cross layer design protocols is presented at the end of the section.

2. MAC protocols for WSN

In designing a MAC protocol for a Wireless Sensor Network (WSN), some of the unique features of WSN must be taken into consideration. Low-power consumption must be the main goal of the protocol. The coordination and synchronization between nodes must be minimized in the protocol. The MAC protocol must be able to support a large number of nodes. It must have a high degree of scalability. The MAC protocol must take into account the limited bandwidth availability. Since sensor nodes of a WSN are deployed randomly without a predefined infrastructure, the first objective of the MAC protocol for a WSN is the

creation of the network infrastructure. The second objective is to share the medium communication between the sensor nodes (Ian et al. 2002).

IEEE 802.11 is a well-known MAC protocol for Ad hoc network (IEEE working group 1999). The energy constraints in the sensor nodes make it is unpractical to apply the IEEE 802.11 protocol directly in WSN. IEEE 802.11 has a power save mode. The power save mode in IEEE 802.11 is designed for a single hop network, where all nodes can hear each other. This is not the case in WSN. A set of MAC protocols for the WSN were proposed. Most of the existing protocols aimed to save power consumption in the sensor nodes. In the following subsections, we will discuss most of MAC protocols for WSN

2.1 S-MAC protocol

The main goal of S-MAC is to reduce energy consumption while supporting good scalability and collision avoidance. (Wei et al. 2004) extend PAMAS (Sureh S. and Cauligi 1998) by using a single channel for transmitting data packets and control packets. In designing S-MAC protocol they assume that WSN composed of many small nodes deployed in an Ad Hoc fashion. Moreover they assume that most communication will be between nodes as peers rather than one base station. It is assumed that the sensor nodes are self configured and the sensor network is dedicated to a single application or a few collaborative applications. The sensor network has the ability of in-network processing.

Ye et al identify four sources for energy wasting. The first source is collisions which will cause retransmission the packet. Transmission will consume power. The second source is overhearing; picking a packet intended to another node. The third source of energy consumption is transmission of control packets. The final source of energy consumption is idle listening. S-MAC reduces the energy waste due to these reasons. The basic idea of S-MAC is to let the node sleep and listen periodically. In sleeping mode, the node turns its radio off. The listening period is fixed according to physical layer and MAC layer parameters. The complete cycle of listening and sleeping periods is called a frame. The duty cycle is defined as the ratio of the listening interval to the frame length. Neighboring nodes can be scheduled to listen and sleep at the same time. Two neighboring nodes may have different schedules if they are synchronized by different two nodes. Nodes exchange their schedule by broadcasting a SYNC packet to their immediate neighbors. The period to send a SYNC packet is called the synchronization period. If a node wishes to transmit a packet to its neighbor it must wait until its neighbor becomes in its listening period. Fig. 1 shows 4 neighboring nodes A, B, C, and D. Nodes A and C are synchronized together (they have the same schedule , they listen and they sleep at the same time) while nodes B and D are synchronized together.

Fig. 1. S-MAC: Neighboring nodes A and B have different schedules. They synchronize with nodes C and D respectively

S-MAC forms nodes into a flat, peer-to-peer topology. To choose a schedule the node firstly listens for a fixed amount of time (at least the synchronization period). If the node does not receive a schedule within the synchronization period, the node chooses its own schedule and starts to follow it, and then it announces its schedule to its neighbors by broadcasting

the SYNC packet. If it hears a schedule from one of its neighbors before it chooses or announces its own schedule, it follows that schedule. If a node receives a different schedule after it announces its own schedule, then there will be two cases, in the first case, the node has not other neighbors, then it discard its own schedule and it will follow the new schedule. In the second case, the node already follows a schedule with one of its neighbors; therefore it will adopt both schedules by waking up at the listening intervals of the two schedules. To maintain the schedule, each node maintains a schedule table that stores the schedules of all its known neighbors. To prevent case two in which neighbors miss each other forever when they follow two different schedules, a periodic neighbor discovery is introduced. Each node periodically listens for the whole synchronization period. If multiple nodes wish to talk to the same node that is in listening period, then all of them must contend for the medium. IEEE 802.11 scheme with RTS and CTS is used to avoid collision, which will save energy consumption due to the packets collision and retransmissions.

To avoid overhearing which is one of the sources of energy consumptions, each interfering nodes must go to sleep after they hear RTS and CTS. All immediate neighbors of both sender and receiver should sleep after they hear RTS or CTS. To reduce the delay due to sleeping, a technique called adaptive listening is integrated in S-MAC. Each node will wake up for a short period at the end of the transmission. In this way, if the node is the next-hop node, its neighbor is able to pass the data immediately to it instead of waiting for its scheduled listening time.

To reduce energy consumed due to control packet overhead, a message passing technique is included in S-MAC. If a node wishes to transmit a long message, the long message is fragmented into fragments and the node will transmit them in burst; one RTS and one CTS are used for all the fragments. When a node sends data, it waits for ACK. The ACK is useful to solve the hidden terminal problem. Data fragment and ACK packets have a duration field. If a node wakes up or joins the network and it receives a data or ACK packet, it will go to sleep for the period in the duration field in data or ACK packet.

Synchronization among neighboring nodes is required to remedy their clock drift. Synchronization is achieved by making all nodes exchange a relative timestamps and letting the listening period is longer than clock drift.

A disadvantage of S-MAC is that the listening interval is fixed regardless whether the node has data to send or there are data intended to it. a Traffic Aware, Energy Efficient MAC protocol is proposed for WSN (TEEM) (Chansu & Young-Bae 2005). They extend the S-MAC protocol by reducing the listening interval.

2.2 A Traffic Aware, Energy Efficient MAC protocol for Wireless Sensor Networks (TEEM)

The TEEM protocol is an extension to S-MAC In S-MAC protocol the listening interval is fixed while in TEEM protocol the listening interval depends on the traffic. In TEEM protocol; all nodes will turn their radio off much earlier when no data packet transfer exists. Furthermore, the transmission of a separate RTS is eliminated. In TEEM protocol; each listening interval is divided into two parts instead of three parts as in S-MAC protocol. In the first part of the listening interval, the node sends a SYNC packet when it has any data message ($SYNC_{data}$). If the node has no data message, it will send a SYNC packet ($SYNC_{nodata}$) in the second part of its listening interval. $SYNC_{data}$ is combined with RTS packet to form $SYNC_{rts}$. If a node does not receive $SYNC_{data}$ in the first part of its listening

interval and it has no data to send it will send $SYNC_{nodata}$ in the second part of its listening interval. If a node receives a $SYNC_{rts}$ that is intended to another node, it will turn its radio off and goes to sleep until its successive listening interval starts. The intended receiver will send CTS in the second part of its listening interval. The performance evaluation of TEEM protocol shows that the percentage of sleeping time in TEEM is greater than the percentage of sleeping time in S-MAC. The number of control packets in TEEM is less than the number of control packets in S-MAC. Energy consumption in TEEM is the least compared with S-MAC and IEEE 802.11. Although the power consumption is reduced in the TEEM by decreasing the listening interval, the latency will increase since decreasing the listening interval depends only on the local traffic, traffic in the node itself and in the neighboring node, and does not take into account the traffic in the whole network. To take into account the delay in the whole network, Lin et al propose a sensor medium access control protocol with a dynamic duty cycle, DSMAC (Peng et al. 2004). DSMAC intend to achieve a good tradeoff between power consumption and latency.

2.3 Medium ACCES Control with a Dynamic duty cycle for sensor network (DSMAC)

In S-MAC the duty cycle is fixed. In DSMAC the duty cycle is changed based on average delay of the data packet and the power consumption (Peng et al. 2004). The duty cycle is defined as the ratio of the listening interval to the frame length; the frame length is the sleeping interval plus the listening interval. Duty cycle can be changed by changing the sleeping interval while fixing listening interval. As in S-MAC, the nodes in DSMAC form groups of peers. Each set of neighbors follow a common schedule. In DSMAC, one- hop packet latency is proposed which is the time since a packet gets into the queue until it is successfully sent out. The packet latency is recorded in the packet header and sent to the receiver. The receiver calculates the average packet latency. The average packet latency is an estimation of the current traffic. If the average packet latency is larger than a threshold delay (D_{max}), and if the energy consumption level greater than a threshold energy (E_{max}), then the duty cycle will be doubled by decreasing the sleeping interval such that the new frame length is half of the original frame length. Otherwise the duty cycle will be halved by doubling the sleeping interval, doubling the sleeping interval will double frame length. The purpose of changing the duty cycle by two (or half) is to maintain the old schedule, which enables neighboring nodes to communicate using the old schedule.

2.4 Timeout-MAC (T-MAC)

In T-MAC, the node will keep listening and transmitting as long as it is in an active period (Tijs & Koen 2003). An active period ends when no activation event has occurred for a specific time TA. An activation event may be firing of a periodic frame timer, reception of any data on the radio, sensing of communication on the radio, end-of-transmission of a node's own data packet or acknowledgement, or the knowledge that a data exchange of a neighbor has ended. Communications between nodes in T-MAC is performed using RTS/CTS mechanism. The node that wishes to transmit data must send an RTS and wait for the CTS. If it does not receive CTS within the TA period the node will go to sleep. The node does not receive CTS in three cases; the receiver has not received the RTS, the receiver receives RTS but it is prohibited from replying, or the receiver is sleeping. It is accepted and recommended for the node to go to sleeping in the third case. But it is not an optimal

decision to go to sleeping in the first two cases. To take into account all the three cases; when the node does not receive CTS to the first RTS it will resend another RTS and if it does not receive a response to the second RTS then it will go to sleeping. Sending two RTS packets without getting a CTS indicates that the receiver cannot reply now so it is convenient for the sender to go to sleeping. TA must be long enough to receive at least the start of the CTS packet. Overhearing avoidance is achieved by the same technique used in S-MAC. One problem of the T-MAC is the early sleeping problem, which occurs in case of asymmetric communication where there are four consecutive nodes: *A*, *B*, *C*, and *D*. node *A* sends data to *B* which its final destination is *C*, at the same time *C* wishes to send data to node *D* but it cannot transmit data since a collision will occur at node *B* with the transmission form *A* to *B*, so node *C* will go to sleeping. Moreover, node *D* will go to sleeping. Later when node *B* wishes to forward the data to node *C*, it will find that node *C* is sleeping which will make node *B* to go to sleeping and transmit its data later which will increase the delay and decrease the throughput. Two solutions are proposed: future request-to-send and taking priority on full buffers (Tijs & Koen 2003).

2.5 GANGS Protocol

There are some applications, in which most of the traffic in the nodes is a forwarding traffic. For these network models, Biaz et al propose a MAC protocol (GANGS) in which the nodes are organized into clusters 0(Saad & Yawen 2004). The communication within the cluster is contention based and the communication between cluster heads is TDMA based. GANGS is an energy efficient MAC protocol. As the other protocols, the nodes in GANGS are organized into clusters. Each cluster has a head. The heads form the backbone of the sensor network. The communication between nodes within cluster is contention based while the communication between heads is TDMA based. The frame is divided into multiple contention free TDMA slots and one contention slot. Number of TDMA slots depends on the number of neighboring clusters heads. The radios of all normal nodes will be turned OFF through TDMA slots while the radios of all heads are turned ON through the entire frame.

Establishing the cluster consists of three stages: local maximum stage, inter-cluster stage and reconfiguration stage. In the local maximum stage, the nodes communicate with their neighbors and exchange their energy information. The node that has the local maximum energy claims that it is the head and sends this claim to its neighbors. In the Inter-cluster phase, new heads are added to construct the backbone. Any node that it is not a head may be in the range of one head and accepts it as a head, in the range of multiple heads and it needs to choose one of them, or it is not in the range of any head. If it is in the range of multiple heads and if it has a maximum energy, then it will be the new head, otherwise the node will select the head with the maximum power. If it is not in the range of any head, then it sends a message to a node with local maximum power to demand head service. The node with local maximum power will be the new head. Since the head consumes more energy, eventually it will no longer have the maximum energy and reconfiguration must be performed to select new heads.

As any TDMA based protocol, Synchronization between the cluster heads is needed. To arrange the TDMA schedule each head knows number of its neighbors, each head randomly choose a number in the range [1, number of neighbors+1]. Each head sends the chosen number to its neighbors. If the chosen number is the same, the head with less number of

neighbors will change its schedule. All the nodes will synchronize themselves with the head to which they belong to it.

3. Routing Protocols for WSN

WSN has distinguished characteristics over traditional wireless network that makes routing in WSN is very challenging. First; it is not possible to build a global addressing scheme due to the deployment of huge number of sensor nodes, therefore the classical IP-based routing protocols cannot be applied to sensor networks. Second, Most applications of the sensor networks require the data flow from multiple sources to a particular sink. Third, the generated data has significant traffic redundancy in it. Furthermore, sensor nodes have limited power resource and processing capacity. Due to such differences many routing protocols for WSN are proposed. The routing protocols are classified as data centric, hierarchical, or location based (Kemal & Mohamed 2005). Data-centric protocols are query-based and depend on naming of desired data. Hierarchical protocols aim at clustering the nodes so that cluster heads can do some aggregation and reduction of data to reduce energy. Location based protocols utilize the position information to relay data to the desired region rather than the whole network.

Flooding is a classical mechanism to relay data in sensor network without using any routing protocol. In flooding, each sensor node receives a data packet; it will broadcast data to all its neighbors (Sandra & Stephen 1988). Eventually the data packet will reach its destination. To reduce the data traffic in the network, gossiping is implemented in which a receiving node send packet to a randomly selected neighbors. In flooding and gossiping, a lot of energy is wasted due to unnecessary transmissions. In addition to energy loss, flooding and gossiping have many drawbacks such as implosion where duplicated message sent to the same node, and overlap where many nodes sense the same region and send similar packets to the same neighbors.

3.1 Data-Centric protocols

In data-centric routing protocol, the sink sends queries to specific regions and the sensor nodes located in the selected region will send the corresponding data to the sink (Kemal & Mohamed 2005)0. To specify the properties of the requested data, attribute-based naming is usually used. Many data centric routing protocols are proposed.

Directed Diffusion: In Directed Diffusion, a naming scheme for the data is used; attribute-value pairs for the data are used (Chalermek C. et al. 2000). The sensor nodes are queried on demand using attribute-value pairs. To create a query, an interest is defined using a list of attribute-value pairs such as name of objects, interval, duration and geographical area. The interest is broadcasted by the sink. Each node receives the interest will cache it along with the reply link to a neighbor from which the interest is received. The reply link which is called a gradient is characterized by data rate, duration and expiration time. To establish the path between the sink and source, each node will compare the attribute of received data with the values in the cached interest. Using the gradients, the receiving node will specify the outgoing link. Path repairs are possible in Directed Diffusion, when a path between a source and sink fails, a new path should be identified. Multiple paths are identified in advances so that when a path fails one of the alternative paths is chosen without any cost of searching for another path. Directed Diffusion has many advantages; since all

communication is neighbor-to-neighbor there is no need for addressing mechanism. Using caching will reduce processing delay. Moreover, Direct Diffusion is energy efficient since the transmission is on demand and there is no need for maintaining global network topology. On the other hand, directed diffusion can not be applied to all sensor network-application since it is based on query-driven data delivery model. It can not be used for applications that require continues data delivery such as environmental monitoring. In addition, the data naming   scheme used in Directed Diffusion is application dependent, it must be defined in advance.

Rumor Routing: Rumor Routing (David & Deborah 2002) is another variation of the Directed Diffusion. It is based on a query-driven data delivery model. In Rumor Routing, the queries are routed only to the nodes that have observed a particular event instead of querying the entire network as in Directed Diffusion. In Rumor Routing, each node maintains a list of neighbors and events table with forwarding information to all the events it knows. When a node senses an event, it adds it to its event table with a distance of zero to the event, and it generates an agent. An agent is a long-lived packet that travels the network in order to propagate information about local events to all the nodes. The agent contains an events table similar to the table in the nodes. Any node may generate a query for an event; if the node has a route to the event, it will transmit the query. If it does not, it will forward the query in a random direction. This continues until the query TTL expires, or until the query reaches a node that has observed the target event. If the node that originated the query determines that the query did not reach a destination it can retransmit or flood the query.

A New Gradient Based Routing Protocol: (Li et al. 2005) proposes a new gradient-based routing protocol. The proposed protocol takes into account the minimum hop count and remaining energy of each node while relaying data from source node to the sink. The optimal routes can be established autonomously with the proposed protocol. A simple acknowledgement scheme, which is implemented without extra overheads, is proposed. Data aggregation is performed to save transmission energy. To handle the frequent change of the topology of the network, a scheme for frequent change of the topology of the network is provided.

O(1)-Reception Routing Protocol: (Abdelmalik et al. 2007) proposes a technique that enables the best route selection based on exactly one message reception. It is called O(1)-reception . In O(1)-reception, each node delays forwarding of routing messages (RREQs) for an interval inversely proportional to its residual energy. This energy-delay mapping technique makes it possible to enhance an existing min-delay routing protocol into an energy-aware routing that maximizes the lifetime of sensor networks. They also identify comparative elements that help to perform a thorough posteriori comparison of the mapping functions in terms of the route selection precision. The O(1)-reception routing enhances the basic diffusion routing scheme by delaying the interests forwarding for an interval inversely proportional to the residual energy: nodes compute a forwarding delay based on their residual energy and defer the forwarding of interest messages for this period of time. As maximum lifetime routing should combine the min and the max–min metrics, in the energy-delay mapping function, nodes with high residual-energy forward interests without delay to make diffusion equivalent to the min energy routing, and nodes with low residual-energy delay forwarding of interests for a time interval to make diffusion equivalent to the max–min residual energy routing.

Energy-Balancing Multipath Routing (EMPR): The basic idea of EMBR is that the base station finds multipath to the source of the data and selects one of them for data transmission (Yunfeng & Nidal 2006). The base station dynamically updates the available energy of each node along the path based on the amount of packets being sent and received. The base station then uses the updated energy condition to periodically select a new path from multiple paths. The base station takes the role of the server and all sensor nodes work as clients. Base station does every thing from querying specific sensing data, broadcasting control packets, routing path selection and maintenance to work as the interface to the outside networks. Sensor nodes are only responsible for sensing data and forwarding packets to the base station. Topology construction is initiated by the base station at any time. The base station broadcasts Neighbor Discovery (ND) packet to the whole network. Upon receiving this packet, every node records the address of the last hop from which it receives and stores it in the neighbors list in ascending order of receiving time. The node changes the source address of the packet to itself. Then it broadcasts the packet. If the new packet is already received the node drops the ND packet and does not rebroadcast. After the completion of Neighbors discovery, the base station broadcasts another packet, Neighbors collection (NC) to collect the neighbor information of each node. Upon receiving the NC packet, the node replies a NCR (Neighbors Collection Reply) packet by flooding. The base station now has a vision of the topology of the networks through the neighbor's information of all nodes. After the topology construction, the base station constructs a weighted directed graph. The weight of each edge is the available energy of the head node. In the data transmission phase, the base station broadcasts enquiry (DE) for sensing data with specific features. Then the sensor nodes satisfying an enquiry will reply with Data Enquiry Reply (DER) packet. On the other hand, the sensor node does not satisfy the enquiry will rebroadcast DE. The base station calculates the shortest path to the desired node in the weighted node.

3.2 Hierarchical Protocols

In hierarchical routing protocols, clusters are formed. For each cluster, a head node is assigned dynamically, a set of nodes will attach the head node, and the head nodes can communicate with the sink either directly or through upper level of heads. Data aggregation is usually performed at each head.

Low-Energy Adaptive Clustering Hierarchy LEACH: (Wendi et al 2002) propose a LEACH. In LEACH, the nodes organize themselves into clusters. In designing the LEACH, it is assumed that all the nodes in the network can transmit with enough power to reach the base station (BS) of the network and each node has sufficient computational power to support different MAC protocols and perform signal processing functions. Regarding the network model it is assumed that the network consists of nodes that always have data to send to the end user and the nodes which are located close to each other have correlated data.

In LEACH, the nodes organize themselves into local clusters. One of the nodes is identified as a cluster head and all other nodes in the cluster send their data to the cluster head. The cluster head is responsible for processing the data received from the nodes and transmit the resulted data to the base station. Since the cluster head performs data processing and transmission, it will consume more power than normal nodes. The cluster head must be changed through the system life time. Each node must take its turn to act as a cluster head. Operation of LEACH is divided into rounds. Each round begins with a set-up phase

followed by a steady-state phase. In set-up phase, the clusters are formed and the cluster head is assigned. In the steady state phase, the nodes will transmit their data. The algorithm to select a cluster head is a distributed algorithm. Each node makes autonomous decision to be a cluster head. During each round, there are k clusters so there must be k heads. At round $r+1$ which starts at time t, each node selects itself to be a cluster head with probability $P_i(t)$. $P_i(t)$ is chosen such that the expected value of the cluster head must be k. To ensure that all nodes will act as cluster head equal number of times, each node must be a cluster head once in N/k rounds. In (Windy et al 2002) a new probability is proposed to take into account the energy in each node

After identifying the clusters heads, each node must determine the cluster to which it belongs. Each cluster head broadcasts advertisement message containing the head's id using non-persistent CSMA scheme. Each node determines its cluster by selecting the head whose advertise signal is the strongest signal. This head is the closest head to the node. The node will transmit a joint request message to the chosen cluster head using CSMA. Upon receiving all the joint request messages the cluster head sets up the TDMA schedule and transmit this schedule to the nodes in the cluster. Each node will turn OFF its radio all the time slots except their assigned slots. This will end up the set-up phase and start the steady state phase.

The steady state phase is divided into frames; each node sends its data to the cluster head once per frame during its assigned slot. All nodes must be synchronized and start their set-up phase at the same time. This can be done by transmitting a synchronization pulse by the base station to all nodes. To reduce energy dissipation each non head node use power control to set the least amount of energy in the transmitted signal to the base station based on the received strength of the cluster head advertisement. When a cluster head receives the data from all nodes, it performs data aggregation and the resultant data will be sent to the base station. Processing the data locally within the cluster reduces the data to be sent to the base station; therefore the consumed energy will reduced. This is an advantage of the LEACH. To reduce inter-cluster interference, each cluster communicates using direct sequence spread spectrum DSSS. Each cluster uses a unique spreading code.

The distributed cluster formulation algorithm does not offer guarantee about placement and number of cluster head nodes. An alternative algorithm is a central cluster formation; base station (BS) cluster formation. The central cluster formation produce better clusters by dispersing the cluster head nodes throughout the network. In the central algorithm, each node sends information about its current location and its energy level to the BS. The BS computes the average energy level. Any node has energy level less than the average cannot be a cluster head, other nodes can be clusters heads. The BS use simulated annealing to find the cluster heads. The solution must minimize the amount of energy for non-cluster head and find k the optimal number of clusters k_{opt}. When the cluster heads and associated clusters are found the BS broadcasts a message that contains the cluster head ID for each node. (Windy et al 2002) propose a formula to find the optimum number of clusters that minimize the total consumed energy

The frame size in LEACH is fixed regardless of the active nodes in the cluster since it is assumed that all nodes have data to send. This is not the real case all the time, sometimes some of the nodes are active and other nodes are not active.

Energy-Aware Data-Centric Routing Algorithm (EAD): (Azziddine et al. 2005) propose EAD. EAD is designed for event driven application. In EAD, a tree rooted at the base

station is constructed. The tree consists of leaf and non-leaf nodes. A non-leaf node is a node that has at least one child. On the other hand, a leaf node is a node that has no child. All the leaf nodes of the tree will turn their radio OFF most of the time. On the other hand, all the non-leaf nodes will turn their radio ON all the time. When an event occurs, the leaf nodes will collect the related data and turn its radio ON to transmit the data to its parent. When a non-leaf node receives data from all its children, it will aggregate the data and send it to its parent. All the nodes use CSMA/CA for transmitting the data. Since the radio of the non-leaf sensor nodes will always be ON, they will lose much power than the leaf nodes. The tree will be reconstructed from time to time. (Azziddine et al. 2005) proposes an energy aware algorithm to build the tree. One of the disadvantages of EAD is that the non-leaf nodes will be awake all the time even though there are not events to detect. This makes EAD unsuitable for applications with periodic data traffic.

To build a tree rooted at the sink, the sink initiates the process of building the tree. Building the tree is performed by broadcasting control messages. Each control message consists of four fields: *type, level, parent, power*. For the sender node v , $type_v$ represents its status; 0: undefined; 1: leaf node; 2: non-leaf node. $level_v$ refers to the number of hops from v to the sink. $parent_v$ is the next hop of v in the path to the sink; $power_v$ is the residual power E_v. Initially each node has status 0. The sink broadcasts *msg(2,0,NULL,∞)*. When a node *v* receives *msg(2 , $level_u$, $parent_u$, E_u)* from node *u* , it becomes a leaf node, sense the channel until it is idle, then waits for T_2^v time , if the channel is still idle, *v* broadcasts *msg(1 , $level_u$ +1 , u , E_v)*. If *v* receives *msg(1 , $level_u$, $parent_u$, E_u)* from *u* , it senses the channel until it is idle, waits for T_1^v if the channel is still idle , *v* broadcasts *msg(2 , $level_u$ +1 , u , E_v)*. And it becomes non-leaf node. If node *v* receives more than one message from different nodes before it broadcasts its message, it will select the node with larger energy as its parent. If both nodes have the same energy, it will select one of them randomly. The waiting node will go back to sensing state, if another node occupies the common channel before it times out. If a node *v* with status 1 receives *msg(2 , $level_w$, v , E_w)* from node *w* indicating that *v* is its parent, *v* broadcasts *msg(2 , $level_v$, $parent_v$, E_v)* immediately after the channel is idle. The process will continue until each node becomes leaf or non-leaf node. A sensor with status 2 becomes a leaf node if it detects that it has no children. Both T_1^v and T_2^v are chosen such that no two neighboring broadcasts are scheduled at the same time. On the other hand, to force the neighboring sensors with higher energy to broadcast earlier than those nodes with a lower residual power, both T_1^v and T_2^v must be monotonically decreasing functions of E_v. One of the disadvantages of EAD is that all the nodes are connected to the sink through few nodes that are close to the sink. These nodes are considered as gateways. These nodes will be non-leaf nodes for most of time; they will consume a lot of energy. Therefore, they will die early. When they die, the rest of the nodes will be isolated. However, those isolated nodes still have non-consumed energy. Therefore, energy utilization is not so efficient in EAD. (Tayseer & Baroudi 2007) generalize EAD such that any node can act as a gateway.

A Generalized Energy-Aware Data Centric Routing For Wireless Sensor Network (EAD$_{General}$): (Tayseer & Uthman 2007) generalize EAD such that any node can act as a gateway. To generalize EAD, they assume that each node has the ability to transmit its data for long distance, i.e. its transmission can reach the sink. Each node has power control capability such that the transmission energy depends on the distance to the destination node. When a node sends data to its nearest neighbor, the transmission energy will be small

compared with the transmission energy required to transmit data to the sink. In EAD$_{General}$, a new phase; Selecting Gateways (SG), is added. In this phase, gateway nodes are selected. It is assumed that the network is virtually divided into tiers. Each tier includes all nodes that can hear a signal transmitted with specific energy from the sink. For example, $tier_0$ includes all nodes that can hear the signal transmitted from sink with transmission energy equals to E_0. $Tier_1$ includes all nodes that can hear the signal transmitted from sink with transmission energy equals to E_1, where $E_1>E_0$ and so on. Initially, the nodes of $tier_0$ will be considered as potential candidate gateways. Based on their energy level, some of these nodes will advertise themselves as gateways. They will act as gateways until their residual energy drops below a threshold value E_{th}. Then new gateways will be selected from the nodes of $tier_1$. The selected nodes will act as gateways until their residual energy drops below E_{th} and so on. When all tiers are considered and no more nodes can be selected as gateways based on the current E_{th}, a new cycle will start, in this cycle new gateways will be selected from $tier_0$ using smaller value of E_{th} and so on. To select the gateways, the sink broadcasts an *ADV* message. The *ADV* message contains a field for E_{th}. Initially *ADV* message is broadcasted with energy E_0 such that it reaches the nodes of $tier_0$ only. When a node receives the *ADV* message, it compares its residual energy with E_{th}, and then it responds with a *JOIN* message. A *JOIN* message contains a confirmation field. *Confirmation* is set to 1, if the node's residual energy is greater than E_{th}, i.e. the node can be a gateway and it selects the sink as its parent, otherwise *confirmation* is set to 0. After the node sends its *JOIN* message, it will act as gateway in the current round. Assuming reliable channel, it does not need a confirmation from the sink to be a gateway. All nodes send JOIN message with *confirmation field=1* will be considered gateways. If the sink receives *JOIN* messages from all nodes in the target tier and the *confirmation field =0* in all the received *JOIN* messages, then no node from the target tier can be a gateway, since we assume that all nodes can reach the sink, the sink will broadcast a new *ADV* message with higher transmission energy E_1 using the same E_{th} to select a gateway from the next tier. The nodes of the next tier will respond with *JOIN* messages according to their energy. The process will continue until all tiers are considered and no node has energy greater than E_{th}; no node can be a gateway. A new cycle will start from $tier_0$ with new E_{th}, $E_{th}(new)=eE_{th}(current)$, where $0<e<1$. Following the same procedure as above, new gateway nodes will be selected from $tier_0$. For each cycle, a fixed E_{th} will be used, and at the beginning of each new cycle, E_{th} will be reduced by the factor e. The sink and nodes will exchange messages using the CSMA mechanism. The node has to be ON until it receives the *ADV* message from the sink and then it sends the *JOIN* message. Since the node does not need confirmation from the sink, it will go to sleep immediately after sending the *JOIN* message.

A Generalized Energy-Efficient Time-Based Communication Protocol For Wireless Sensor Networks (GET): GET is proposed by (Tayseer & Uthman 2009). In designing GET, they assume that each node has the ability to transmit its data for long distance, i.e. its transmission can reach the sink. Each node has power control capability such that the transmission energy depends on the distance to the destination node. When a node sends data to its nearest neighbor, the transmission energy will be small compared to the transmission energy required to transmit data to the sink. they assumed that all nodes are synchronized. Regarding the application of the network, they assume that the event that is being monitored is periodic, so data transmission from sensor nodes to the sink will start at

specific time, and it will be repeated periodically. They assume also that all the nodes that are located close to each other and have correlated data. Hence, data aggregation will be used and it will reduce data redundancy. In GET, time is divided into rounds. Each round consists of four phases: Selecting the Gateways (SG), Building the Tree (BT), Building the Schedule (BS), and Data Transmission (DT). In the first phase, gateways are selected; the gateway is selected using the algorithm proposed in (Tayseer and Uthman 2007). In the second phase, a tree rooted at the sink is built. The tree is built using building tree algorthim proposed by (Azziddine et al. 2005). They modify the buiding tree algorithm such that building tree process will be initiated by the gatewyas not by the sink Based on this tree, a TDMA schedule is built in a distributed manner in phase-3. The schedule will be built assuming that in the data transmission period, all nodes connected to the sink through the same gateway will use the same frequency to transmit their data.. For each node, they identify two time constants: Time Ready to Receive (*TRR*) and Time Ready to Transmit (*TRT*). For a node *v*, TRR_v represents the time slot when the node is ready to receive data from its children, while TRT_v represents the time slot when a node can transmit data to its parent. Assuming t_0 represents the time at which the periodic sensing event occurred and the data is already collected from the monitored environment. For a leaf node, $TRT_v = t_0$. TRR_v is not valid since it does not have children. On the other hand, for a non-leaf node *v*:

$$TRR_v = Max(TRT_i) \qquad i = 1, 2, 3, \dots n_v^c$$
$$TRT_v = TRR_v + n_v^c T_t \tag{1}$$

Where *i* represent an index for the child of node *v*, , n_v^c represents the count of *v's* children, and T_t represents the time needed to transmit one data packet. To build the schedule, initially, each leaf node will transmit its *TRT* value to its parent. When a parent receives *TRT* values from all its children, it calculates its *TRR* and *TRT* using (1) and builds the schedule for its children. Then it transmits its *TRT* to its parent and broadcasts the schedule to its children. The process will continue until all nodes receive their assigned time slot from their parents. Both leaf and non-leaf nodes use *CSMA/CA* protocol to exchange data (*TRT* and the Schedule). Eventually, we have a TDMA schedule for the whole sensor network.

In the fourth phase, data is transmitted from sensor nodes to the sink following the schedule prepared in phase-3. Data transmission period represents the time needed to forward all data packets in a single round. Data transmission period may be repeated many times in a single round

TinyDB: Another alternative in the same direction is the work presented in (Samuel et al. 2005). A distributed query processor for smart sensor devices (TinyDB) is proposed. In TinyDB, to disseminate queries and collecting results, a routing tree rooted at the base station is built. The routing tree is formed by forwarding a routing request (a query in TinyDB) from every node in the network. The root sends a request then all child nodes that hear this request process it and forward it on to their children, and so on, until the entire network has heard the request. Each node picks a parent node that is one level closer to the root. This parent will be responsible for forwarding the node's query results to the base station. To limit the scope of queries, a Semantic Rooting Tree (SRT) is built. This tree is built based on the routing tree. If a node knows that none of its children currently satisfies the query, it will not forward the query down the routing tree. Therefore, each node must have information about child attribute values.

Unequal Cluster Based routing (UCR): In UCR protocol, clusters with different size are constructed (Guihai et al. 2007). Cluster heads closer to the sink will have smaller cluster sizes than those farther from the sink. Thus they can preserve some energy for the inter-cluster data forwarding. A greedy geographic and energy-aware routing protocol is designed for the inter cluster communication which considers the tradeoff between the energy cost of relay paths and the residual energy of relay nodes. The UCR protocol consists of two parts: an energy-efficient unequal clustering algorithm called EEUC and an intercluster greedy geographic and energy-aware routing protocol. Initially, the base station broadcasts a beacon signal to all sensors at a fixed power level. Based on the received signal strength, each sensor node can compute the approximate distance to the base station. It not only helps nodes to select the proper power level to communicate with the base station, but also helps us to produce clusters of unequal sizes. In EEUC algorithm, heads will be identified randomly. As in LEACH protocol, the task of being a cluster head is rotated among sensors in each round to distribute the energy consumption across the network. After cluster heads have been selected, each cluster head broadcasts a CH_ADV_MSG across the network field. Each ordinary node chooses its closest cluster head, the head with the largest received signal strength, and then informs it by sending a JOIN_CLUSTER_MSG. After forming clusters, data will be transmitted from the cluster heads to the base station. Each cluster head first aggregates the data from its cluster members, and then sends the packet to the base station via a multi-hop path through other intermediate cluster heads. Before selecting the next hop node, each cluster head broadcasts a short beacon message across the network at a fixed power which consists of its node ID, residual energy, and distance to the base station. A threshold TD_MAX in the multi-hop routing protocol is proposed. If a node's distance to the base station is smaller than TD_MAX, it transmits its data to the base station directly; otherwise, it is better to find a relay node that can forward its data to the base station.

Energy-aware routing for cluster-based sensor networks: (Younis et al. 2002) proposed a hierarchical routing algorithm based on a three-tier architecture. In the proposed protocol, sensors are grouped into clusters. The cluster heads (gateways) are less energy constrained than normal sensors. It is assumed that cluster heads knows the location of the sensor nodes. Gateways maintain the states of the sensors and sets up multi-hop routes for collecting sensors data. Each gateway informs each node within its clusters the time slots in which it can transmit and in which it have to listen to other nodes transmission. The sensor nodes in the cluster can be in one of four states: sensing only, relaying only, sensing-relaying and inactive. In sensing state the sensor node senses the environment and generates the corresponding data. In the relaying only state, the node does not sense the environment but it forwards data from other active nodes. In sensing-relaying state, the node not only senses the environment but also forwards the data from other active nodes. In inactive state, the node neither senses the environment nor forwards data. The link cost is defined as the energy consumption to transmit data between two nodes, the delay optimization and the other performance cost. A least-cost path is found between sensor nodes and the gateway. The gateway monitors the available energy level at every sensor that is active. Rerouting is triggered by an application-related event requiring different set of sensors to probe the environment or the depletion of the battery of an active node.

Base-Station Controlled Dynamic Clustering Protocol (BCDCP): (Muruganathan et al. 2005) proposes a clustering-based routing protocol called Base Station Controlled dynamic

Clustering protocol (BCDCP). In BCDCP, the base station sets up clusters and routing paths, performs randomized rotation of cluster heads, and carries other energy intensive tasks. The key ideas in BCDCP are: formulation of balanced clusters where each cluster head serves an approximately equal number of member nodes, uniform placement of cluster heads throughout the entire sensor field, and the utilization of cluster-head-to-cluster-head(CH-to-CH) routing to transfer the data to the base station. Class-based addressing of the form <Location ID, Node Type ID> is used in BCDCP. The Location ID identifies the location of a node. It is assumed that the base station keeps up-to-date information on the location of all the nodes in the network. A Node Type ID describes the functionality of the sensor such as seismic sensing, and thermal sensing. BCDCP operates in two major phases: setup and data communication. In setup phase, clusters are formed, clusters' heads are selected, CH-to-CH routing paths are formed, and schedule is created for each cluster. During each setup phase, the base station receives information on the current energy status from all the nodes in the network. Based on this information, the base station computes the average energy level and then chooses a set of nodes, denoted S, whose energy levels are above the average value. Cluster heads for the current round will be chosen from the set S. To identify the cluster heads from the set and to from clusters, iterative cluster splitting algorithm is used. This simple algorithm first splits the network into two sub-clusters, and proceeds further by splitting the sub-clusters into smaller clusters. The base station repeats the cluster splitting process until the desired number of clusters is attained. Once the clusters and the cluster head nodes have been identified, the base station chooses the lowest-energy routing path and forwards this information to the sensor nodes along with the details on cluster groupings and selected cluster heads. The routing paths are selected by connecting all the cluster head nodes using the minimum spanning tree approach that minimizes the energy consumption and then a head is randomly selected to transmit data to the base station. The last step in this phase is building a TDMA Schedule for each cluster. In The data communication phase, Data gathering, Data fusion, and Data routing is performed using the TDMA schedule created in setup phase.

3.3 Location-Based Protocols

Information Location can be utilized to forward data with minimum energy consumption. If the region to be monitored is known, the query can be forwarded to that region. Many location-based routing protocols for WSN were proposed. In the successive subsections, I will survey many of these protocols.

Geographic Adaptive Fidelity (GAF): GAF is energy-aware location-based routing protocol designed for mobile ad hoc protocols, but it can be applicable to sensor networks (Ya et al. 2001) . In GAF a virtual grid for the monitored area is formed. Each node uses its GPS-indicated location to associate itself with a point in the virtual grid. Nodes associated with same point in the grid are equivalent. Some of them can be in the sleeping state to save energy while others will be in active state. Therefore, the network lifetime will increase. To balance load among nodes, equivalent nodes change their state from active to sleeping in turn. Three states are defined in GAF, discovery, sleep, and active. In the discovery state a node will determine its neighbors. While it is in sleep state, a node will turn OFF its radio. The active node will participate in data routing. A node will be in each state for particular time period which is application dependent. On the other hand, determining which nodes that will be in sleep state is application dependent. GAF is implemented for non-mobility

(GAF-basic) and mobility (GAF-mobility adaptation) of nodes. To keep the network connected, a representative node must be always active for each region on its virtual grid.

Geographic and Energy Aware Routing (GEAR): In GEAR protocol, energy aware and geographical-informed neighbor selection heuristic is used to route packets towards the destination region (Yan et al. 2001). The key idea is to restrict the number of interests in Directed Diffusion to certain regions rather than sending interest to the whole network. Each node keeps an estimated cost and a learning cost of reaching the destination through its neighbors. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a refinement of the estimated cost. A hole exists in the network when a node does not have any closer neighbor to the target region. With no holes in the network, the estimated cost is equal to the learned cost. When a packet reaches the destination, the learned cost is propagated one hop back so that route setup for next packet will be adjusted. The GEAR protocol consists of two phases; in the first phase, the packets are forwarded towards the target region, when a node receives a packet, it checks its neighbors to see if there is a neighbor that is closer to the target region. The closest neighbor to the target region is selected as the next hop. When all neighbors are further than node itself, a hole exists; one of them will be selected based on the learned cost function. This selection will be updated according to the convergence of the learned cost. In the second phase, packets will be forwarded within the region; the packets are forwarded in the region by either recursive geographic forwarding or restricted flooding.

A Mesh-Based Routing Protocol for Wireless Ad-Hoc Sensor Network (MBR): In MBR protocol, the area of the sensor network is portioned into regions; mesh topology (Foad & r, Hadi 2006). The nodes can communicate to their neighbor nodes through virtual channels. Forming the mesh topology is performed in three phases. In the first phase, the base node for zoning is selected. Two setup sensors are determined. One of them is located at the largest diameter and in the boundary of the area and the second sensor is located on the boundary of other orthogonal diameter of the region. In phase two, the network is divided into regions. In phase three, each sensor nodes is assigned ID. Each sensor will be known with two features: its region coordinate (X,Y) and its ID. To transmit data between source nodes and sink a path is reserved between them firstly. To reserve a path, the source node sends a reserve message, called RAP, to the sensors in its target (X,Y). Upon receiving the RAP message, each node generates a priority number and returns it to the source node using ACK message. Sensors have higher energy will have higher priority. The source sensor will select sensors to form the path among the sensors that sends ACK message. Then data will be sent based on the path determined. After transmitting data, path must be released. This is done by sending a CRP message.

Energy-efficient geographic multicast routing: (Juan et al. 2007) proposes a novel energy-efficient multicast routing protocol called GMREE. It aims to preserve energy and network bandwidth. GMREE protocol builds multicast trees based on a greedy algorithm using local information. GMREE protocol is based in the concept of cost over progress metric and it is specially designed to minimize the total energy used by the multicast tree. The cost is defined as the energy needed to reach the furthest neighbor in the selected set of relays plus the energy that such amount of nodes will need to process the message. GMREE incorporates a relay selection function which selects nodes from a node's neighborhood taking into account not only the minimization of the energy but also the number of relays selected. Nodes only select relays based on a locally built and energy-efficient underlying

graph reduction such as Gabriel graph, enclosure graph or a local shortest path tree. Thus, the topology of the resulting multicast trees really takes advantage of the benefit of sending a single message to multiple destinations through the relays which provide best energy paths.

Energy-Aware Geographic Routing for Sensor Networks with Randomly Shifted Anchors: Anchor-based geographic routing aims at finding a small number of intermediate nodes acting as anchors so that the path length (i.e. number of hops) between the source and destination can be reduced. However, some nodes (e.g., nodes near the boundary of the network) tend to be used as anchors repeatedly by multiple flows. As a result, their energy drains quickly and the lifetime of the network is reduced. Moreover, the intermediate nodes between source and destination change very little once the anchor list is set. This also contributes to the quick depletion of the energy for some nodes. To overcome these shortcomings, (Gang et al. 2007) introduces a random shift to the location of each anchor in the routing process. Each new packet will then be routed to a different anchor determined by the location of the original anchor plus the random shift. Because the shift is generated randomly, different packets will likely be routed through a different list of anchors. This allows more nodes to be involved in the routing process and the energy consumption is better distributed among nodes in the network.

On Optimal Geographic Routing in Wireless Networks with Holes and Non-Uniform Traffic: Subramanian et al. propose a randomized geographic routing scheme that can achieve a throughput capacity of $\Theta(1/\sqrt{n})$ (within a poly-logarithmic factor) even in networks with routing holes (Sundar et al. 2007). They show that the proposed scheme is throughput optimal (up to a poly-logarithmic factor) while preserving the inherent advantages of geographic routing. They also show that the routing delay incurred by the proposed scheme is within a poly-logarithmic factor of the optimal throughput-delay trade-off curve. On the other hand, Subramanian et al. construct a geographic forwarding based routing scheme that can support wide variations in the traffic requirements as much as $\Theta(1)$ rates for some nodes, while supporting $\Theta(1/\sqrt{n})$ for others. They show that the above two schemes can be combined to support non-uniform traffic demands in networks with holes.

The randomized algorithm takes as input the number of nodes in the network, the packet to be sent, as well as the number of holes. Considering the first packet in all the source nodes, The source node for every traffic flow creates Rlog(n) copies of its packet to send. It chooses Rlog(n) independent and uniformly distributed points from the unit region and sets the NEXT-DEST field in the packet to the randomly generated location in each of these copies. The Rlog(n) packets are routed from the source in a greedy geographic manner to the location in NEXTDEST. Upon receiving a packet, a node checks if it is the NEXTDEST location. If it is not the NEXT-DEST location, it searches within its neighboring nodes for the node that is closest to the NEXT-DEST location, and forwards the packet to that node. If none of its neighbor nodes is closer to the NEXT-DEST than itself, the node drops the packet. If it is the NEXT-DEST location, it checks whether it is the final destination or not. If it is the final destination, then the packet is received. Otherwise, If the final destination is one hop away from the current node, the node forwards the packet greedily to the final destination. If the final destination is more than one hop a way from the current node, the current node makes Rlog(n) copies of the packet and again generates uniform and randomly

chosen locations for the NEXT-DEST in each of the packet copies, and forwards them greedily.

3.4 QoS-aware Protocols

QoS-aware protocols consider end-to-end QOS requirement while setting up the paths in the sensor network. Many QoS-aware routing protocols for WSN were proposed. In the successive subsections, I will survey many of these protocols.

Maximum Lifetime Energy Routing: (Jae-Hwan et al. 2000) presents a routing protocol for sensor networks based on a network flow approach. The protocol aims to maximize the network lifetime by defining link cost as a function of node remaining energy and the required transmission energy using that link. Finding traffic distribution is a possible solution to the routing problem. The solution to this problem maximize the network lifetime. Two maximum residual energy path algorithms were proposed to find the best link metric for the maximization problem. The two algorithms differ in their definition of link costs and the incorporation of nodes' residual energy. The least cost paths to destination are found using Bellman-Ford shortest path algorithm. The least cost path is the path whose residual energy is largest among all paths.

Maximum Life Time Data Gathering: (Konstantinos et al. 2002) models the data routes setup in sensor network as the maximum lifetime data-gathering problem. A polynomial time algorithm to solve this problem is proposed. The data-gathering schedule specifies for each round how to get and route data to sink. For each round, a schedule has one tree rooted at the sink and spans all the nodes of the network. The network lifetime depends on the duration for which the schedule remains valid. The Maximum Lifetime Data Aggregation (MLDA) protocol is proposed to set up maximum lifetime routes taking into account data aggregation. If a schedule "S" with "T" rounds is considered, it induces a flow network G. the flow network with maximum lifetime subject to the energy constraints of sensor nodes is called an optimal admissible flow network. A schedule will be constructed by using this admissible flow network. For application with no data aggregation such as video sensors, a new scenario is presented, which is called Maximum Lifetime Data Routing (MLDR). It is modeled as a network flow problem with energy constraints on sensors.

SPEED: SPEED is a real-time communication protocol for sensor networks (Tian et al. 2003). It provides three types of real-time communication services; real-time unicast, real-time area-multicast and real-time area-anycast. SPEED is a stateless, localized algorithm with minimal control overhead. End-to-end soft real-time communication is achieved by maintaining a desired delivery speed across the sensor network through a novel combination of feedback control and non-deterministic geographic forwarding. SPEED is a highly efficient and scalable protocol for sensor networks where the resources of each node are scarce. In SPEED protocol, each node should maintain information about its neighbors. Geographic forwarding is used to find the paths. SPEED protocol strives to ensure end-to-end delay for the packets in the network such that each application can estimate the end-to-end delay for the packets. SPEED protocol consists of the following components: A neighbor beacon exchange scheme, a delay estimation scheme, The Stateless Non-deterministic Geographic Forwarding algorithm (SNGF), A Neighborhood Feedback Loop (NFL), Backpressure Rerouting, and Last mile processing. SNGF is the routing module responsible for choosing the next hop candidate that can support the desired delivery speed. NFL and Backpressure Rerouting are two modules to reduce or divert traffic when congestion occurs,

so that SNGF has available candidates to choose from. The last mile process is provided to support the three communication semantics mentioned before. Delay estimation is the mechanism by which a node determines whether or not congestion has occurred. And beacon exchange provides geographic location of the neighbors so that SNGF can do geographic based routing. Table 1 shows a classification of routing protocols based on the application.

Protocol	Application		
	Query Based	Event Driven	Periodic
SPIN	√		
Directed Diffusion	√		
Shah et al.			√
Rumor Routing	√		
CADR	√		
COUGAR	√		
ACQIRE	√		
GBR	√		
O(1)-Reception Routing Protocol		√	
EMPR	√		
LEACH		√	
EAD		√	
TinyDB	√		
PEGASIS		√	
TEEN		√	
APTEEN			√
UCR		√	
BCDCP		√	
GAF		√	√
MECN		√	
GEAR	√		
GOAFR		√	
MBR		√	
GMREE		√	
Zhao et al. Randomly Shifted Anchors:		√	
Chang et al			√
Kalpakis et al.		√	√
Minimum Cost Forwarding		√	
SAR		√	√
Energy-Aware QoS Routing Protocol			√
EAD$_{General}$		√	
SPEED	√		
GET			√

Table 1. Classification of Routing Protocols based on the Applications

4. Literature Review of Cross Layer design in WSN

Many researchers studied the necessity and possibility of taking advantages of cross layer design to improve the power efficiency and system throughput of Wireless sensor network. (Safwat et al. 2003) proposed Optimal Cross-Layer Designs for Energy-efficient Wireless Ad hoc and Sensor Networks . They propose Energy-Constrained Path Selection (ECPS) scheme and Energy-Efficient Load Assignment (E2LA). ECPS is a novel energy-efficient scheme for wireless ad hoc and sensor networks. it utilizes cross-layer interactions between the network layer and MAC sublayer. The main objective of the ECPS is to maximize the probability of sending a packet to its destination in at most n transmissions. To achieve this objective, ECPS employs probabilistic dynamic programming (PDP) techniques assigning a unit reward if the favorable event (reaching the destination in n or less transmissions) occurs, and assigns no reward otherwise. Maximizing the expected reward is equivalent to maximizing the probability that the packet reaches the destination in at most n transmissions. Ahmed Safwat et. al, find the probability of success at an intermediate node i right before the tth transmission $f_t(i)$:

$$f_t(i) = \begin{cases} 1 & i = D \\ \max_j \sum_k p_k f_{t+k}(j) & otherwise \end{cases}$$

(2)

where D is the destination node and j is the next hop towards the destination D. Any energy-aware route that contains D and the distance between D and the source node is less or equal to n can be used as input to ECPS. The MAC sub-layer provides the network layer with the information pertaining to successfully receiving CTS or an ACK frame, or failure to receive one. Then ECPS chooses the route that will minimize the probability of error

The objective of the E2LA scheme is to distribute the routing load among a set Z of Energy-aware routes. Packets are allotted to routes based on their willing to save energy. Similar to ECPS, E2LA employs probabilistic dynamic programming techniques and utilize cross-layer interactions between the network and MAC layers. At the MAC layer, each node computes the probability of successfully transmitting packets in α attempt. E2LA assign loads according to four distinct reward schemes (Safwat et al. 2003).

(Venkitasubramaniam et al. 2003) propose a novel distribution medium access control scheme called opportunistic ALOHA (O-ALOHA) for reachback in sensor network with mobile agent. The proposed scheme based on the principle of cross layer design that integrates physical layer characteristics with medium access control. In the O-ALOHA scheme, each sensor node transmits its information with a probability that is a function of its channel state (propagation channel gain). This function called transmission control is then designed assuming that orthogonal CDMA is employed to transmit information. In designing the O-ALOHA scheme they consider a network with n sensors communicate with a mobile agent over a common channel. It is assumed that all the sensor nodes have data to transmit when the mobile agent is in the vicinity of the network. Time is slotted into intervals with equal length equal to the time required to transmit a packet. The network is assumed to operate in time division duplex (TDD) mode. At the beginning of each slot, the collection agent transmits a beacon. The beacon is used by each sensor to estimate the propagation channel gain from the collection agent to it which is the same as the channel gain from the sensor to the collection agent. It is assumed that the channel estimation is

perfect. The propagation channel gain from sensor *i* to the collection agent during slot *t* which is

$$\gamma_i^{(t)} = \frac{P_T R_{it}^2}{r_i^2 + d^2}$$

(3)

Where R_{it}^2 : is Rayleigh Distribution, and P_T is the transmission power of each sensor, and r_i is the radial distance of sensor i , and d is the distance from collecting agent and sensor node. During the data transmission period, each sensor transmits its information with a probability $S(\gamma_i^{(t)})$ where $S(.)$ is a function that maps the channel state to a probability. Two transmission controls are proposed to map from the channel gain to the probability; Location independent transmission control (LIT) and Location aware transmission control (LAT). In LIT, the decision to transmit a packet is made by observing channel state γ alone, while in LAT, every sensor makes an estimate of its radial distance and the decision to transmit is a function of both the channel state γ and the location of sensor.

(Sichitiu 2004) proposed a deterministic schedule based energy conservation scheme. In the proposed approach, time synchronized sensors form on-off schedules that enable the sensors to be awake only when necessary. The energy conservation is achieved by making the sensor node go to sleeping mode. The proposed approach is suitable for periodic applications only, where data are generated periodically at deterministic time. The proposed approach requires the cooperation of both the routing and MAC layers. The on-off schedule is built according to the route determined by routing protocol. The proposed approach consists of two phases; the Setup and reconfiguration phase and the steady state phase. In the setup and reconfiguration phase, a route is selected from the node originating the flow to the base station then the schedules are setup along the chosen route. In the steady phase, the nodes use the schedule established in the setup and configuration phase to forward the data to the base station. In this phase, there will be three types of actions at each node; Sample action which is taking data sample from environment, Transmit action to transmit data, and Receive action to receive data. The actions at each node along with the time when each action will take place are stored in the schedule table of each node. The node can be awake ate the time of each action and go to sleep otherwise.

(Li-Chun & Chung-Wei 2004) proposed Cross layer Design of Clustering architecture for wireless Sensor Networks. The proposed scheme is called Power On With Elected Rotation (POWER). The objective of the POWER is to determine the optimal number of clusters from the cross-layer aspects of power saving and coverage performance simultaneously. The basic concept of the POWER is to select a representation sensor node in each cluster to transmit the sensing information in the coverage area of the sensor node. The representative sensor node in a cluster rotated from all the sensor nodes in each cluster. In the POWER scheme, the scheduling procedure is rotated many rounds. In each round, there are two phases; the construction table phase (CTP), to construct the rotation table and the rotational representative phase (RRP) to transmit data. In CTP, all sensor nodes employ the MAC protocol and the first sensor node accessing the channel become the initiator node, then the initiator node detects other neighboring node and form s the cluster. RRP starts after constructing the rotation table. RRP is divided into many sRPs (Sub-Rotated Period). In each sRP, one node will be a representative node and all other nodes in the cluster will be in sleeping mode.

Protocol	Layers	Approach	Evaluation method	Applica-tion	Network Topology	Cross layer Objective	Performance metrics
ECPS	MAC, Network	Mathematical Model: probabilistic dynamic programming	Experiment		Random (Static)	Maximization of probability of sending packet to its D at n transmission	Energy
E2LA	MAC, Network	Mathematical Model: probabilistic dynamic programming	Experiment		Random (Static)	Minimize Energy:- Multiple simultaneous routes Load distribution	Energy
MAC CROSS	MAC, Network	Heuristic	Simulation Hardware Implemen-tation (MICAZ)		Random (Static)	Maximize Sleep Duration	Energy
O-Aloha	Physical, MAC	Heuristic	Simulation	SENMA	Random	Maximize throughput	Throughput
POWER	Physical, MAC, Network	Heuristic			Uniform (Static)	Optimize number of cluster	Energy
Weilian Su	ALL layers	Framework (optimization Agent)	Experiment al (MICAZ)		Random	Optimize performance of WSN	Link Quality Packet Received
Shunguang Cui	Routing, MAC, Link layer	Modeling as optimization problem	Analytical		Random	Maximize network lifetime	Network lifetime
Sense-Sleep Trees (SS-Trees)	MAC, Network	Heuristic	Simulation	Surveill-ance	Mesh-based	Maximizing Network lifetime, and monitoring coverage	Network lifetime Energy consumed
Game Theoretic Approach	Applicat ion, Physical	Game Theory	Analytical		Random	Minimize total distortion	Distortion coverage
In Yeup Kong	Physical, MAC, Network	Mathematical	Analytical		Random	Maximize Network lifetime	
Cross Layer Scheduling	MAC, Network	Heuristic	Simulation	Periodic	Random	Maximize network lifetime	Network lifetime
Cross Layer design for cluster formulate-on	MAC, Physical, Network	Heuristic	Simulation	Periodic	Uniform distribution	Maximize network lifetime	Network lifetime

Table 2. Summary of Cross layer Protocols for W

(Rick et al. 2005) proposes a cross-layer sleep-scheduling-based organization approach, called Sense-Sleep Trees (SS-trees). The proposed approach aims to harmonize the various engineering issues and provides a method of increasing monitoring coverage and operational lifetime of mesh-based WSNs engaged in wide-area surveillance applications. An iterative algorithm is suggested to determine the feasible SS-tree structure. All the SS trees are rooted at the sink. Based on the computed SS-trees, optimal sleep schedules and traffic engineering measures can be devised to balance sensing requirements, network communication constraints, and energy efficiency. For channel access a simple single-channel CSMA MAC with implicit acknowledgements (IACKs) is selected. In SS-trees approach, the WSN's life cycle goes through many stages. After the initial deployment of nodes, the WSN will enter the network initialization stage, in which the sink gathers network connectivity information from sensor nodes, compute the SS-trees, and disseminate

the sleep schedules to every sensor node. Then the WSN will enter the operation stage, in which the nodes will alternate between Active and sleep stages. During long periods when sensing services are not needed the entire WSN will enter the Hibernation mode to conserve energy. The SS-trees must be computed with minimizing number of shared nodes (nodes belonging to multiple SS-trees), minimizing co-SS tree neighbors of each node, and minimizing the cost of forwarding messages between the data sink and each node. Rick W. Ha et al proposes a greedy algorithm to compute the SS-trees. The proposed algorithm follows a greedy depth-first approach that constructs the SS-trees from the bottom up on a branch-by-branch basis. After computing the SS-trees, an optimal sleep schedule that maximizes energy efficiency must be determined. The length of the active and sleep period will increase the data delay. The proposed SS-Tree design streamlines the routing procedures by restricting individual sensor nodes to only maintain local connectivity information of its immediate 1-hop neighbors.

(Shuguang et al. 2005) emphasize that the energy efficiency must be supported across all layers of the protocol stack through a cross-layer design. They analyze energy-efficient joint routing, scheduling, and link adaptation strategies that maximize the network lifetime. They propose variable-length TDMA schemes where the slot length is optimally assigned according to the routing requirement while minimizing the energy consumption across the network. They show that the optimization problems can be transferred into or approximated by convex problems that can be solved using known techniques. They show that link adaptation be able to further improve the energy efficiency when jointly designed with MAC and routing. In addition to reduce energy consumption, Link adaptation may reduce transmission time in relay nodes by using higher constellation sizes such as the extra circuit energy consumption is reduced.

(Weilian and Tat 2006) propose a cross layer design and optimization framework, and the concept of using an optimization agent (OA) to provide the exchange and control of information between the various protocol layers to improve performance in wireless sensor network. The architecture of the proposed framework consists of a proposed optimization agent (OA) which facilitates interaction between various protocol layers by serving as a database where essential information such as node identification number, hop count, energy level, and link status are maintained. (Weilian and Tat 2006) conduct the performance measurements to study the effects of interference and transmission range for a group of wireless sensors. The results of their performance measurements help to facilitate the design and development of the OA. The OA can be used to trigger an increase in transmit power to overcome the effects of mobility or channel impairments due to fading when it detects a degradation due in BER. Alternatively, it can reduce the transmit power to conserve energy to prolong its lifetime operations in the absence of mobility or channel fading. The OA can also be used to provide QoS provisioning for different types of traffic. This can be done by tagging different priority traffic with different transmit power levels.

(Changsu et al. 2006) proposed an energy efficient cross-layer MAC protocol for WSN. It is named MAC-CROSS. In the proposed protocol, the routing information at the network layer is utilized for the MAC layer such that it can maximize sleep duration of each node. in MAC-CROSS protocol the nodes are categorized into three types: Communicating Parties (CP) which refers to any node currently participating in the actual data transmission, Upcoming Communicating Parties (UP) which refers to any node to be involved in the actual data transmission, and Third Parties (TP) which refers to any node are not included

on a routing path. The UP nodes are asked to wake up while other TP nodes can remain in their sleep modes. The RTS/CTS control frames are modified in the MAC-CROSS protocol. The modification is needed to inform a node that its state is changed to UP or TP in the corresponding listen/sleep period. a new field; Final_destination_Addr, is added to the RTS. On the other hand, a new field; UP_Addr is added to the CTS and it informs which node is UP to its neighbors. When a node B receives an RTS from another node A including the final destination address of the sink, B's routing agent refers to the routing table for getting the UP (node C) and informs back to its own MAC. The MAC agent of Node B then transmits CTS packet including the UP information. After receiving the CTS packets from node B, C changes its state to UP and another neighbor nodes change their states to TP and will go to sleep.

Table 2 shows summary of cross-layer design protocols for WSN.

5. Conclusion

In this chapter, we present a summary for MAC, Routing, and Cross layer Design protocols for WSN. In section 0, a survey of MAC protocols for WSN is presented. The routing protocols for WSN are discussed in section 0. A classification of the routing protocols according to the application is presented in section 3. Section 0 presents a summary of cross layer design protocols for WSN. A summary of cross layer design protocols at the end of section 4.
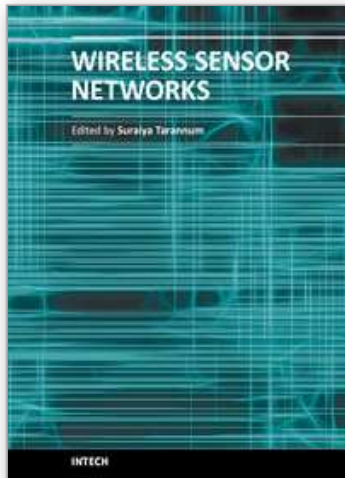
6. References

Ian F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002). A survey on sensor networks. *IEEE Personal Communications Magazine*, August.

The working group for WLAN standards 1999). IEEE 802.11 standards, Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications. *Technical report*, IEEE

Sureh S. and Cauligi S. Raghavendra 1998), "PAMAS: Power aware multi-access protocol with signalling for ad hoc networks," *ACM Comput. Commun. Rev.*, vol. 28, no. 3, July 1998, pp. 5–26,.

Wei Ye, John Heidemann, and Deborah Estrin, Fellow 2004), "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, Vol 12, No. 3, June 2004, pp. 493-506,.

Chansu Suh, Young-Bae Ko. (2005), "A traffic Aware, Energy Efficient MAC Protocol for Wireless Sensor Networks", *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005.* pp.2975 - 2978 Vol. 3 , 23-26 May 2005

Peng Lin, Chunming Qiao and Xin Wang (2004) "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," in WCNC, Mar 2004.

Tijs van Dam and Koen Langendoen (2003), "An adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in ACM Sensys'03, Nov. 2003.

Saad Biaz, Yawen Dai Barowski (2004), "GANGS: an Energy Efficient MAC Protocol for Sensor Networks", ACMSE'04 April 2-3.

Kemal Akkaya and Mohamed Younis (2005), "A Survey of Routing Protocols in Wireless Sensor Networks, " in *the Elsevier Ad Hoc Network Journal*, 2005 vol. 3/3 pp. 325-349.

Sandra. M. Hedetniemi and Stephen. T. Hedetniemi (1988), "A survey of gossiping and broadcasting in communication networks," Networks, Vol. 18, No. 4, 1988, pp. 319-349,

Chalermek Intanagonwiwat, Ramesh Govindan and Deborah Estrin (2000), "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.

David Braginsky and Deborah Estrin (2002), "Rumor Routing Algorithm for Sensor Networks," in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.

Li Xia, Xi Chen, and Xiaohong Guan Xiac (2005), A New Gradient-Based Routing Protocol in *Wireless Sensor Networks Embedded Software and Systems*, Springer Berlin, Heidelberg, 2005.

Abdelmalik Bachir, Dominique Barthel, Martin Heusse, and Andrzej Duda (2007), "O(1)-Reception routing for sensor networks," *Computer Communications* Volume 30 , Issue 13, (2007), pp. 2603-2614.

Yunfeng Chen, and Nidal Nasser (2006), "Energy-Balancing Multipath Routing Protocol for Wireless Sensor Networks," in the Proc. of the third International Conference on Quality of Service in Heterogeneous Wired/Wireless Network, Waterloo, Ontario, Canada, August 7-9, 2006.

Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan (2002), "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE On Wireless Communications Trans.*, vol. 1, No. 4, Oct. 2002, pp. 660-670.

Azzedine Boukerche, Xuzhen. Cheng, Joseh. Linus (2005), "A Performance Evaluation of a Novel Energy-Aware Data-Centric Routing Algorithm in Wireless Sensor Networks", *Wireless Networks* 11, 2005, pp.619–635,

T. AL-khdour, U. Baroudi (2007), " A Generalized Energy-Aware Data Centric Routing For Wireless Sensor Network", in the Proc. of The 2007 IEEE International Conference on Signal Processing and Communications (ICSPC 2007) , Dubai, United Arab of Emirates (UAE), Nov. 24–27.

T. AL-khdour, U. Baroudi (2009), "A Generalized Energy-Efficient Time-Based Communication Protocol for Wireless Sensor Networks", *Special issue of International Journal of Internet Protocols (IJIPT)*, Vol. 4, No. 2-2009.

Samuel R. Madden Madden, Michael J. Franklin And Joseph M. Hellerstein, And Wei Hong (2005) , "TinyDB: An Acquisitional Query Processing System for Sensor Networks", *ACM Transaction on Database Systems*, Vol. 30, No 1, March 2005, Pages 122-173.

Guihai Chen, Chengfa Li , Mao Ye, and Jie Wu, (2007) "An Unequal Cluster-Based Routing Strategy in Wireless Sensor Networks ," *Wireless Networks* (JS) , April 2007.

Younis M., Youssef M. and Arisha K. (2002), "Energy-Aware Routing in Cluster-Based Sensor Networks", in the Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), Fort Worth, TX, October 2002.

Muruganathan, S.D.; Ma, D.C.F.; Bhasin, R.I.; Fapojuwo, A.O. (2005), "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks," *IEEE Radio Communication,* March 2005, pp. S8-S13.

Ya Xu, John Heidemann, and Deborah Estrin (2001), "Geography-informed energy conservation for ad hoc routing," in the Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, July 2001.

Yan Yu, Ramesh Govindan, and Deborah Estrin (2001), "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.

Foad. Lotfifar, Hadi. Shahhoseini (2006), "A mesh-Based Routing Protocol for Wireless Ad-Hoc Sensor Networks," in the Proc. of International Wireless Communication and Mobile Computing Conference (IWCMC'06), Vancouver, British Columbia, Canda, July 3-6, 2006.

Juan A. Sanchez, Pwdro M. Ruiz, and Ivan Stojmenovic (2007), "Energy-efficient geographic multicast routing for Sensor and Actuator Networks," *Computer Communications* 30 (2007) pp. 2519–2531

Gang Zhao, Xianggian Liu, and Min-Tue Sun (2007), "Energy-Aware Geographic Routing for Sensor Networks with Randomly Shifted Anchors," in the Proc. of Wireless Communications and Networking Conference WCNC 2007, 11-15 March 2007, pp. 3454-3459

Sundar Subramanian, Sanjay Shakkottai and Piyush Gupta (2007), "On Optimal Geographic Routing in Wireless Networks with Holes and Non-Uniform Traffic," in the Proc. of 26th IEEE International Conference on Computer Communications. INFOCOM 2007, May 2007, pp. 1019-1027

Jae-Hwan. Chang, Lendros and Tassiulas (2004), "Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking (TON) archive* Volume 12 , Issue 4 (August 2004) ,pages: 609 - 619

Konstantinos Kalpakis, Koustuv Dasgupta and Parag Namjoshi (2004) , "Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks," in the Proceedings of IEEE International Conference on Networking (NETWORKS '02), Atlanta, GA, August 2002.

Tian He, John A Stankovic, Chenyang Lu, and Tarek Abdelzaher (2003), "SPEED: A stateless protocol for real-time communication in sensor networks," in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.

Safwati. A., Hassanein. H., Mouftah. H. (2003)," Optimal Cross-Layer Designs for Energy-Efficient Wireless Ad hoc and Sensor Networks", in the Proceedings of the IEEE International Conference of Performance, Computing, and Communications 9-11 April 2003 Page(s):123 – 128

Venkitasubramaniam P., Adireddy S., Lang Tong (2003), "Opportunistic ALOHA and cross layer design for sensor networks" , Military Communications Conference, 2003. MILCOM 2003. IEEE Volume 1, 13-16 Oct. 2003 Page(s):705 - 710

Sichitiu M.L. (2004), "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks" ,INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies , Volume 3, 2004 Page(s):1740 - 1750

Li-Chun Wang, Chung-Wei Wang (2004), "A Cross-layer Design of Clustering Architecture for Wireless Sensor Networks", in the Proceedings of the IEEE International Conference on Networking, Sensing & Control Tapel, Taiwan, March 21-23, 2004, Page(s): 547-552

Rick W. Ha, Pin-Han Ho and X. Sherman Shen (2005), "Cross-Layer Application-Specific Wireless Sensor Network Design with Single-Channel CSMA MAC over Sense-Sleep Trees", *Elsevier Journal: Computer Communications Special Issue on Energy Efficient Scheduling and MAC for Sensor Networks*, WPANs,WLANs, and WMANs, 2005

Shuguang Cui, Madan R. , Goldsmith A. , Lall S. (2005), "Joint routing, MAC, and link layer optimization in sensor networks with energy constraints " IEEE International Conference on Communications, ICC 2005 ,Volume 2, 16-20 May 2005 Page(s):725 - 729

Su. W., T.L. Lim (2006), "Cross-Layer Design and Optimization for Wireless Sensor Networks," Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD June 2006 Page(s):278 – 284

Changsu Suh, Young-Bae Ko, and Dong-Min Son (2006) , "An Energy Efficient Cross-Layer MAC Protocol for Wireless Sensor Networks", APWeb 2006, LNCS 3842, pp. 410–419, 2006.

Wireless Sensor Networks

Edited by

ISBN 978-953-307-325-5

Hard cover, 342 pages

Publisher InTech

Published online 29, June, 2011

Published in print edition June, 2011

INTECH
open science | open minds