Beacon Activation MAC Protocol: A hybrid of RIMAC and TMAC Protocol

A Dissertation submitted

By Khushbu Rani

To

Department of Computer Science and Engineering

In criteria for fulfillment to

Award the Degree of

Master in Technology in Computer Science and Engineering

Under the guidance of

Richa Jain

Assistant Professor, LPU

April, 2015

ABSTRACT

The dissertation report on MAC protocol proposes a new MAC protocol named BCAMAC (Beacon Activation MAC), a hybrid of RIMAC and TMAC protocol. The introductory chapter gives an overview of the arena of WSN (Wireless Sensor Network) beginning by showcasing the origin and a brief history of WSN. Further the chapter deals with managing WSN, issues in WSN like efficiency, topology, sensor node deployment etc. Also routing techniques used in WSN is elaborated. Subsequent sections of the chapter detail sensor node types, architecture, functions, and characteristics and so on. Second chapter outlines MAC protocol and related whereabouts and examples. Energy saving mechanisms viz duty cycling, scheduling, on demand wake up, directional antenna and clustering is the plinth of the entire chapter. Other peripheral techniques viz routing, topology control and data aggregation of saving energy are also listed. Further types of MAC protocol based on organization and design approach is explained elaborately. Next chapter titled presents the work done to propose, discuss, compare and validate the new MAC protocol using an apt research methodology. The results and related discussion can be found in the subsequent chapter. The final chapter is the zest of the entire thesis report concluding the work done based on observations as well as enumerates the future scope of the field.

CERTIFICATE

This is to certify that Miss Khushbu Rani, registration no. 11309511 has completed M.Tech dissertation titled "**Beacon Activation MAC Protocol: A hybrid of RIMAC and TMAC Protocol**" under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation is fit for the submission and meets the degree fulfillment conditions for the award of M.Tech in Computer Science and Engineering.

Date**: 05 May, 2015**

Place**: Phagwara (India)**

ADVISOR**:**

Richa Jain

Assistant Professor

UID: 17688

LPU

ACKNOWLEDGEMENT

Enthralled of presenting my M.Tech thesis report on a contemporary subject, I express my gratitude for the innumerous people around me and in my life, who have voluntarily or involuntarily provided me the backbone and plinth to stand upon accomplishing the task. First and the foremost, the credit goes to my family for their unconditional mental, emotional, physical and financial support every moment of my life. I give my respect and many thanks to my mentor for her support, guidance, coordination and perseverance. I must also acknowledge my friends and University management for their presence and discussions as well.

DECLARATION

I Khushbu Rani hereby declare that the dissertation entitled "**Beacon Activation MAC Protocol: A hybrid of RIMAC and TMAC Protocol**" submitted for fulfillment of M.Tech Degree in CSE (Computer Science and Engineering) is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for award of any other degree or diploma to any University or Organization.

Date**: 05 May, 2015**

Place**: Phagwara (India)**

Khushbu Rani

Reg No. 11309511

LPU

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Wireless sensor network (WSN) is a mounting technology of present, that assurances an extraordinary and immense capability to instrument, track, monitor and ultimately control the sensitive tasks of the corporal world remotely and wirelessly to make it worthy. It can be said to be an intelligent distributed measurement technology.WSN consists of numerous wireless sensor nodes heavily spread and networked over the considered region of interest to perform acute tasks of identification, computing, processing, managing and maintenance. Speedy growth of multiple sensing technologies MEMs (Micro Electro Mechanical Systems), VLSI (Very Large Scale Integration) and wireless communication are key contributor of modern WSNs, lending ubiquitous computing and communication for the modern network traffic that is a vicious amalgam of real time traffic such as multimedia, voice , games and data traffic such as messaging, web browsing and file transfer.

Section 1 of this chapter discusses the origin and brief history of WSN. The consequent section gives overview of WSN management, section 3 details sensors, section 4 summarizes WSN features; section 5 discusses issues in WSN.

1.1. ORIGIN AND A BRIEF HISTORY OF WSN

The first wireless network showcasing practical similarities to any present WSN was SOSUS (Sound Surveillance System) developed by US Military. SOSUS served to analyze, track, monitor and control activity of submarines in 1950s. Many initiatives were made in 1960-1970s to create the hardware for Internet we use today. US Defense Research Projects Agency (DARPA) started DSN (Distributed Sensor Silicon Laboratories Network) program in 1980s. DSN explored the issues in implementing various classes of sensor networks. As DSN stepped into academics through partnering universities, WSN technology entered into academia and noncombatant scientific research. Universities and government began implementing WSN in applications such as forest fire detection, air content monitoring etc.

Soon engineering scholars delved into corporate technology giants Bell Labs, IBM and were promoting the use of WSN in heavy industrial application. [1]

Examples of some academic/industrial ingenuities include:

1. UCLA Wireless Integrated Network Sensors (1993)
2. PicoRadio program at University of California in Berkeley (1999)
3. μ Adaptive Multi-domain Power Aware Sensors program at MIT (2000)
4. NASA Sensor Webs (2001)
5. ZigBee Alliance (2002)
6. Center for Embedded Network Sensing (2002).

1.2. SENSOR NODES

Sensors mean to computers what senses mean to human. The core of any type of WSN lies in the sensors. Sensor nodes are actually small autonomous computers boarded on to a sensing board deployed wirelessly in the environment (sensor field), powered by a battery and tied to a backbone network. It is provided with a processor, a memory and a number of modules for sensing such as meteorological, seismic and infrared etc. Sensor nodes can be MEMS (micro-electro-mechanical systems) such as accelerometers, acoustic sensors etc., CMOS (Complementary Metal Oxide Semiconductor) such as temperature, humidity, chemical composition or LED such as proximity sensing, ambient light sensing etc. [3] Applications that can prevail in environments, tolerant to slight variation of temperature and small duration are provided with Lithium Manganese Dioxide ($LiMnO_2$) batteries. Whereas applications that have to undergo large temperature shifts and need to have long lifetime, have deployed sensors equipped with Lithium Thionyl Chloride ($LiSOCl_2$). [1] TinyOS is the embedded OS running on motes. Motes are smart sensor nodes. Nowadays, all sensor nodes are motes. TinyOS is an extremely optimized OS, as it is fast inspite of being tiny. It also supports here and now tasks like multithreaded and event oriented tasks. C variant of TinyOS called nesC is used for programming purposes. NesC presents an event driven concurrency control. [2] Functions of motes include self-diagnosis, reliability, self-identification, software services, standard control protocols (IEEE 1451 Expo, 2001).Wireless sensor nodes communicate via multi-hop network to transmit an amalgam of traffic to the destination node. This destination node purpose could be to gather data for end user retrieval or

transfer of it across a link via intermediate nodes. Data reporting method followed by sensor nodes can be time-driven, event-driven, or query-driven. The first method is used when data transmission is a consequence of time like transmission at constant periodic intervals, second method is applicable when sensor nodes are triggered instantly as a response to one or several event occurrence. The later method is used when sensor nodes respond to a request, query or reply by base-station or any other node. Data reporting method could also be the hybrid of all the above three mentioned methods. A sensor node comprises of a power unit; a sensing unit; a processing unit, and a transceiver as its major components. There are numerous sensing units each consisting of sensor and ADC (Analog to Digital converter). Processing unit has a storage space and processor.

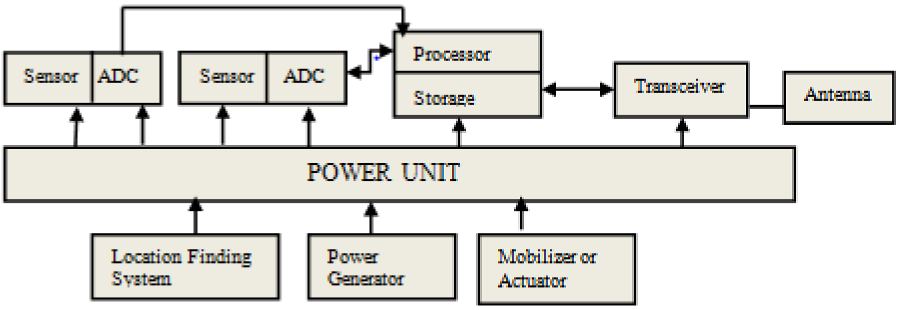

Figure 1: Sensor Node Components

A sensor node is small in dimension consequently imposing restriction on the resources available on the board. For example, low power processor, limited storage space, limited range sensors, scalar sensors, slow data rate etc. The batteries used to power sensors, have some lifetime adding to the menace of inapplicability of being re-charged or replaced due to environmental as well as cost constraints. Therefore energy on board of wirelessly deployed sensors, is a limited resource and needs to be utilized cautiously. Many energy foraging mechanisms exists in our environment to recharge batteries like solar power or wind energy but an optimized mechanism for harnessing them in air is still a wide research area.

Table 1: Sensor Network Platforms and their specifications [36]

			Rene	**Mica2**	**Tmote Sky**	**Imote2**
1	Year		1999	2002	2005	2007
2	CPU		ATMEL 8535	ATmega128L	TI MSP430	Intel PXA271
3	Memory	RAM	512B	4KB	10KB	32MB
		Flash	8KB	128KB	48KB	32MB
4	Radio		RFM TR1000	CC1000	CC2420	CC2420

Table 2: Power characteristics of MICA2 mote sensor

	Radio State	Power Consumption
1	Transmit	81 mW
2	Receive/Idle	30 mw
3	Sleep	0.003 mW

Table 3: Power consumption values of commonly used radios

	RADIO	**PRODUCER**	**TRANSMISSION**	**RECEPTION**
1	CC2420	Texas Instruments	35 mW (at 0 dBm)	38 mW
2	CC1000	Texas Instruments	42 mW (at 0 dBm)	29 mW
3	TR1000	RF Monolithics	36 mW (at 0 dBm)	9mW
4	JN-DS- JN513x	Jennic	111mW (at 1 dBm)	111mW

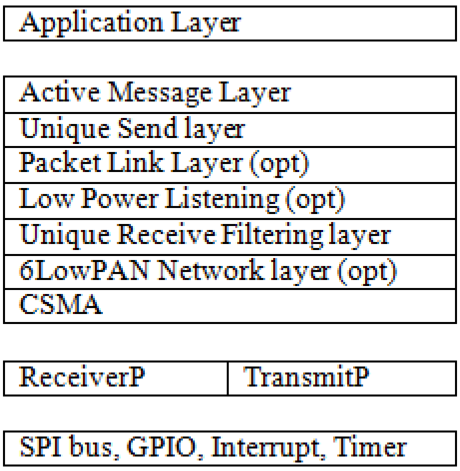

Fig 2: Architectural layers of CC2420 radio stack

Table 4: Power consumption values of some regular sensors [4]

	Sensor	Producer	Sensing	Power Consumption (in mW)
1	STCN75	STM	Temperature	0.4
2	QST108KT6	STM	Touch	7
3	SG-LINK (1000N)	MicroStrain	Strain Gauge	9
4	SG-LINK (350 N)	MicroStrain	Strain Gauge	24
5	jMEMS	ADI	Accelerometer	30
6	2200 Series	GEMS	Pressure	50
7	T150	GEFRAN	Humidity	90
8	LUC-M10	PEPPERL+FUCHS	Level Sensor	300
9	CP18, VL18,GM60, GLV30	VISOLUX	Proximity	350
10	TDA0161	STM	Proximity	420
11	FCS-GL1/2A4-AP8X-H1141	TURCK	Flow Control	1250

Every sensor has a set of functional characteristics such as wakeup latency and break-even cycle. They influence the energy management strategy of the sensor. The time taken by the sensor to generate a precise value when activated for the first time is called wake up latency. The sensor reading value generated before timeout of wakeup latency is invalid. The rate at which the power consumption of a node possessing a power management policy is same as that of node without any power management policy is called break even cycle. [4]

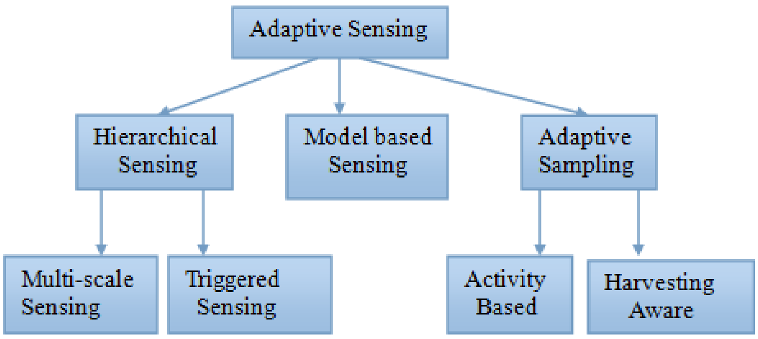

Figure 3: Categories of Adaptive Sensing strategies

In hierarchical sensing technique multiple sensors are mounted on the sensor nodes and they detect the same phenomenon but with a different perseverance and power consumption. In triggered sensing when some activity within the sensed range is detected,

first the low resolution sensor are activated thereafter the more accurate and power consuming sensors ones. In multi-scale sensing within the field being monitored, areas requiring a more precise observation are identified. Adaptive sampling techniques target dynamic adaptation of the sampling rate by manipulating correlations (temporal or spatial) of sensed data with energy pertaining information. Activity-driven adaptive sampling exploits the correlation (both temporal and spatial) among the acquired data. Harvesting-aware adaptive sampling optimizes energy consumption at the unit level by exploiting knowledge about the residual and the forecasted energy coming from the harvester module. Model-based active sampling builds a model of the sensed phenomenon above the primary set of sampled data. Next data can be predicted by using the already built and usable model instead of sampling again. This saves the energy consumed for sensing. The model requires updation or re-estimation, in order to adhere to the dynamicity of the physical phenomenon being observed. A forecasting model is used to build an abstraction of the sensed phenomenon.

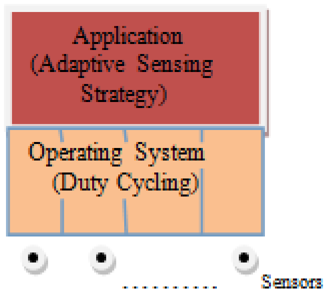

Figure 4: Sensor energy management generalized framework

1.3. WIRELESS SENSOR NETWORKING

WSN components are depicted in Figure 1.2. It comprises of sensor field, sensor node deployed wirelessly, sink node, a task-manager module which could be an application or system, any number of mobile users and backbone of network such as Internet and satellite for connecting the users.

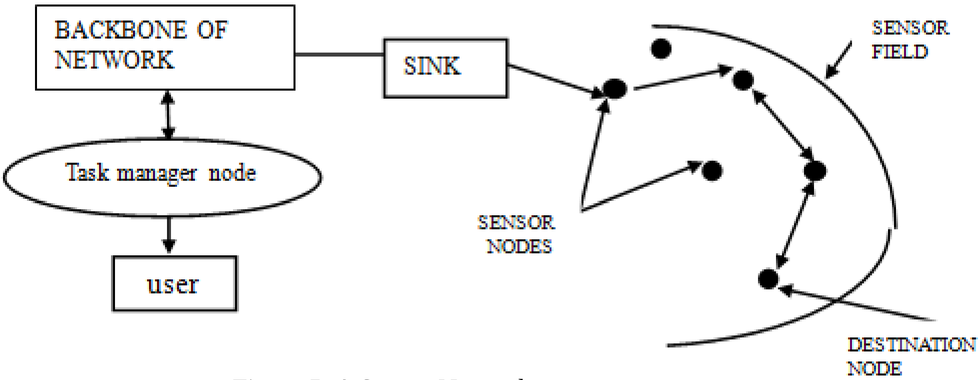

Figure 5: A Sensor Network

Sensor network differ from ad-hoc networks in number of remarkable ways. The count of sensor nodes is more than any ad-hoc network. They are also compactly deployed. Nodes in WSN are disposed to failures as their topology changes very rapidly. WSN topologies can be classified into one way, bidirectional, star and mesh networks. Where most ad-hoc networks are based on communication pattern which is point to point, WSN follows broadcast pattern of communication. Sensor nodes do not possess GID: Global Identification ID because of them being more in number. A sensor network design is influenced by several consequential factors including scalability factor, power consumption, sensor network-topology, hardware restrictions, transmission medium, fault-tolerance, operation environment, production cost. Communication pattern in WSN can be broadcast, local gossip or converge-cast. Broadcast pattern method is used by base-station or a transmitting sender node or a master node to transmit data to all other end and intermediate sensor nodes of the network. Sensors communicate locally in their vicinity, by using local gossip to probe, track, monitor and control any adversaries or intrusion. In converge-cast, a clutch of sensor nodes communicate to a sensor-multicast (a sensor sends a message to a certain subset of sensors). Major causes of energy wastage in WSN are collision, over-hearing, idle-listening, frequent-switching and packet-overhead. Collision is a situation in which a transmitted packet is tarnished due to intrusion or an adversary. In over-hearing, a node has retrieved a packet that was actually meant to be received by some other node. Packet overhead occurs when supporting or associative packets consume maximum amount of energy available and comparatively less or totally unimportant data packets are transmitted. Idle-listening is listening to receive a possible traffic that is not sent at the first place. Frequent switching in transition across various operation modes also results in tremendous energy usage; for instance numerous transitions between active and sleep periods.

1.3.1. Routing techniques in WSN:

Various aspects associated with routing techniques are energy consumption without losing accuracy, data reporting method to be used, network dynamics, node/link heterogeneity, scalability, node deployment, addressing scheme used, fault tolerance, coverage, connectivity, transmission media, data aggregation and QoS. Routing protocols in WSNs may depend on the application (Protocol Operation based) and network architecture (Network Structure based) [2]

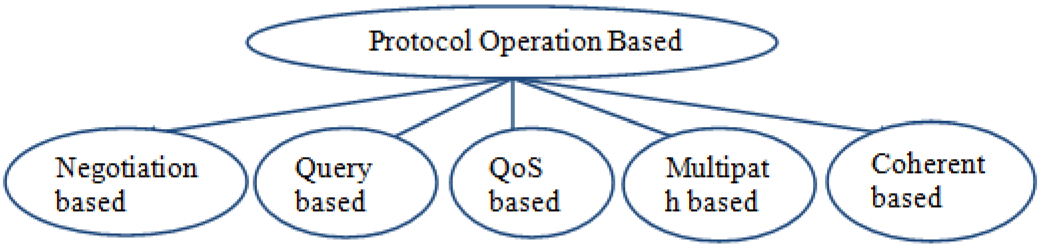

Figure 6: Protocol operation based

Fault tolerance can be enhanced by using multi-path routing protocols. In negotiation based protocols, decisions and transmissions are done based on negotiation with the desired node or network. QoS metrics like delay, bandwidth and energy help balance energy consumed with data quality in QoS based protocol based operation. Coherent based protocols require minimum processing compared to full processing in the other case. In query based routing protocols, destination node transmits a query for data throughout the network. The node having this data replies back with the response to query.

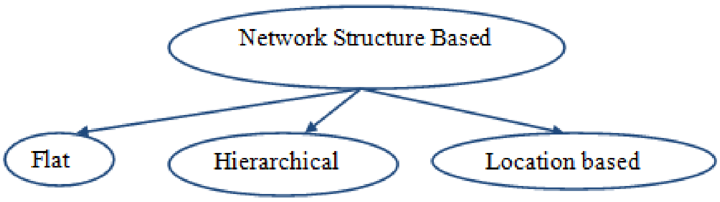

Figure 7: Network structure based

In flat routing protocol, all the nodes playing similar role and all the nodes collaborating to achieve a task. Cluster based routing protocols are also called hierarchical routing protocol. They are characterized by generation of clusters based on the type of task to be accomplished, data aggregation and fusion. Nodes with maximum energy process and transmit information while low power nodes are mainly concerned with sensing the immediate environment. In location based network structure based routing, location of sensor node is important for sensing and communication.

1.3.2. ISSUES IN WSN

Energy consumption in WSN is a major hindrance to the dissemination of WSN technology. In application areas requiring maximum network lifetime and high quality, it proves to be worst. A sensor node is a minute device. This restricts on-the-board critical resources. In fact nodes are charged by batteries with restricted capacity, which cannot be replaced or recharged at convenience or with some cost effective effort due environment and resources constraints. Thus limited resource on-board should be used cautiously. Efficient energy management schemes designed at nodes, at network and cluster level can enhance life of network. Power efficiency in WSNs is generally accomplished in three ways. It could be by low duty-cycle operation, or local/in-network processing or by multi-hop networking. Energy is consumed by a node for sensing, data processing, and communication. Communication consumes the maximum energy. Most power management strategies are based on assumption that data procurement consumes considerably less power than data transmission. Unluckily, this assumption is impractical. There are various issues that influence the fail of the assumption. For example topology issues [37] as illustrated in Figure 8.

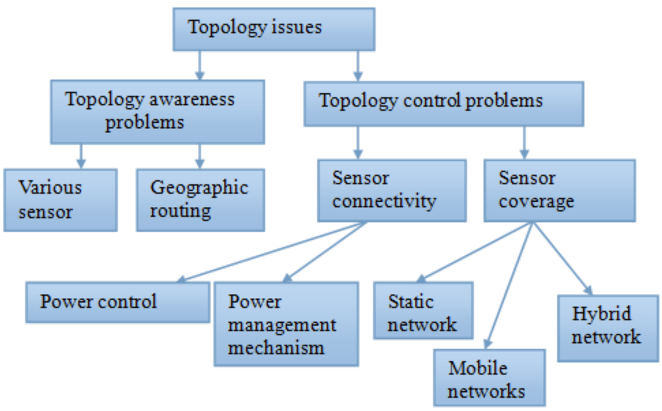

Fig 8: Taxonomy of topology issue in WSNs

1.3.3. WSN MANAGEMENT

WSN can be managed by integrating organizational, administrative and maintenance dimensions as proposed by MANNA architecture. Authors proposed MANNA architecture in an attempt to provide a self-management answer for WSN for temperature monitoring and fire risk evaluation. This paper target is to develop a distributed self-organization service based on policies wherein managers to manager interactions take place (M2M approach) collaborating to achieve a desired goal. The goal can be to form cluster of nodes, control network density, and maintain the WSN area exposure. MANNA defined self-organization as well as self-maintenance policies using the language called PONDER. PONDER is a declarative OOL (Object Oriented Language) used for stating both security and management policies. Policies defined by PONDER are authorization (A) and obligation (O). “A” outlines activities performed by a subject on a set of target objects, while obligation indicated activities to be performed by a manager or an agent, on the target set of objects. Some of the services demarcated by the MANNA architecture are built-in/self: knowledge, awareness, service, organization, configuration, and self-maintenance. A new novel dimension: functionalities on X-axis and other prominent dimensions functional areas on Y-axis and management levels on Z-axis, giving an overview of MANNA architecture is illustrated in Table 1. The management application is distributed into three phases: planning, installation, and operation. The intersection point of the above mentioned three dimensions is called a cell. Each cell consists of a set of management functions. One or several management functions are contained in one or more cells of the cube. Autonomic and centralized management solutions enhance network lifetime. A set of these functions execute “Management services”, and policies established conditions to execute them. The management units exchange management information using MannaNMP. MannaNMP is a simple network management protocol. Management services are executed during operation phase. It is assumed that every node is aware of its own location. Also they are static and make use of omnidirectional antennas. Service negotiation is a dynamic tuning of the network self-service. The service negotiation is epitomised by a state machine. The state machine switching policies define three states: state 0 means event has occurred, state 1 implies risk, state 2 indicates no risk and state 3 in case of an idle environment. [6]

Table 5.WSN Management Axes

Functionalities	Functional Areas	Management Levels
Configuration	Configuration Management	Business
Maintenance	Fault Management	Service
Sensing	Performance Management	Network
Processing	Security Management	Network Element Management
Communication	Accounting Management	Network Element

1.3.4. Applications of WSN: Generally speaking, WSN can be used in application areas where sensor are embedded into systems consequently involving operations like Monitoring, Tracking, Reporting , Collecting, Detecting etc. such wireless sensor networks have also expanded rapidly. Sector wise WSN technology can be used in Agriculture, Environment, Health Care, Military, Industrial, Homeland security, Critical infrastructure protection applications among others. Some examples of specific applications in the above mentioned areas are farmland monitoring, inventory control, smart office, traffic monitoring and control, microclimate control, emergency medical service, farmland monitoring, detection of structural integrity problems, wildlife habitat monitoring, pollution level checks, goods tracking and delivery etc

1.4. MAC (MEDIUM ACCESS CONTROL)

From the perspective of OSI (Open System Interconnection) Reference Model, the MAC protocol functionalities is job of lower sub-layer of the DLL (data link layer). The above sub-layer of the DLL is called as the LLC (logical link control).

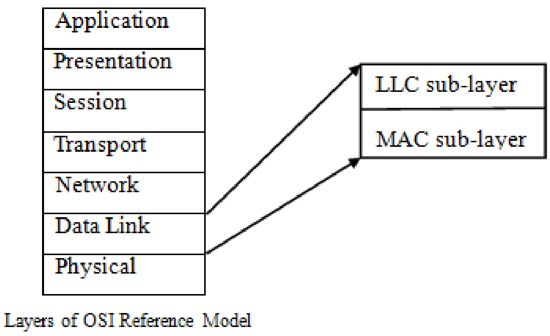

Figure 9: MAC sub-layer

The MAC protocol in a WSN must be multi-hop self-organizing at its best possible and is focused on to achieve two targets. First is creation of the network infrastructure. Second is fair and unbiased sharing of resources among nodes. MAC scheme establishes communication links for data transfer thus forming the basic infrastructure needed by nodes. Because of unavoidable and difficult to handle hardware restrictions, greater energy efficiency relies on energy efficient communicating protocols. An efficient MAC protocol possesses the paramount capability to decrease the energy consumption of the transceiver since it directly controls the operation of it. A MAC protocol provides framing for data packet, medium access control, reliability, flow-control, and error-control. The issues related to MAC protocol can be hidden node or exposed node problem. Performance matrices are used in MAC protocol to gauge the performance of their energy consciousness. Energy consumption per bit, average delivery ratio, latency, and network throughput is used by research community for the purpose. Three modes of MAC operation are DCF: Distributed Coordination Function,

PCF: Point Coordination Function and HCF: Hybrid Coordination Function. MAC protocols have inbuilt power conservation, mobility management and failure recovery strategies. Both fixed-allocation and random-access techniques of medium access have been proposed. Demand based strategies may be inapt for sensor networks as they have large overhead in performing function of messaging and set-up link delayed. Various considerations when designing a MAC protocol that is energy efficient [37] is listed below:

1. Network topology
2. Deployment policy
3. Antenna orientation
4. Control mechanisms
5. Delay
6. Sensing area
7. Transmission range
8. Throughput
9. Quality of Service (QoS) necessities
10. Total number of channels required

There are six major features to demarcate the type of MAC protocol. [37] These are:

1. Topology
2. Channel separation and access
3. Power
4. Transmission initiation
5. Traffic load and scalability
6. Range

1.4.1. Energy saving mechanisms in MAC

There are numerous energy saving mechanisms based on which numerous MAC protocols have been postulated. [37] These are:

1. Duty Cycling
2. Scheduling
3. Scheduled rendezvous
4. On demand wake up scheme
5. Directional antenna
6. Clustering
7. Others: Data rate adaptation, Channel Polling, Hybrid approach.

i. Duty Cycling: In duty cycling approach sensorial system wakes up only for the time needed to attain a novel set of samples after which it is immediately powered off. OS provides a primitives set for the purpose. Later, applications also use these primitives to procure data according to the adaptive sensing strategy implemented. Disadvantages of this strategy is that sensors put into sleep mode may impede working of whole or a part of network. Adaptive duty cycling protocols can help resolve the issue. Synchronization, clock drift are other key issues in duty cycling MAC protocols. Clock drifts can further hamper the performances of MAC protocols if duty cycle and traffic is low.

ii. Scheduling: This mechanism is divided into two categories: Distributed scheduling mechanisms in a non-hierarchical network and in hierarchical networks. Distributed scheduling mechanisms in non-hierarchical networks examples are RIS (Random Independent Scheduling), MSNL (Maximization of Sensor Network Life), LDAS

(Lightweight Deployment Aware Scheduling), PEAS (Probing Environment and Adaptive Sensing), CCP (Coverage Configuration Protocol), PECAS (Probing Environment and Collaborating Adaptive Sleeping), and OGDC (Optimal Geographic Density Control). Examples in case of hierarchical networks include LEACH (Low Energy Adaptive Clustering Hierarchy), E-LEACH (Enhanced LEACH), LDS (Linear Distance-based Scheduling), and BS (Balanced-energy Sleep Scheduling)

iii. Scheduled Rendezvous: Here MAC protocol requires a rendezvous or appointed time at which all neighbouring nodes wake up together. All nodes wake up periodically and sleep until the next rendezvous time. Advantage is that, it is guaranteed and broadcasting is also simple. Disadvantages of such MAC protocol is the requirement of strict synchronization to be maintained as well as clock drift handling.

iv. On Demand Wake Up Scheme: A MAC protocol using this scheme, uses out-of-band radio signals to wake up a dormant node from sleep and embark communication. For the purpose, extra circuit is attached to the sensor board, to wake-up the main radio. A wake-up tone is broadcasted on a separate channel to wake up neighbours. Wake-up tone does not contain any encoded information. It is saves energy since the receiver's job is to detect only energy on the channel and uses a simple hardware further maximizing sleep time for a node. The PicoRadio design uses a low power wake up channel. The project uses a MAC protocol that enables nodes to wake up a neighbour when data needs to be sent. Example: RTWAC. Disadvantage of the scheme is that when using multiple radios, wake-up tone awakes the entire neighbourhood thereby wasting energy in the unnecessarily. Directional antennae can help solve this disadvantage of wake up radio technique by increasing throughput. Although the cost for extra hardware isn't manifold nowadays, the need of an additional hardware for a very large number of sensors can be a disadvantage.

v. Directional Antennae*:* Directional antennae using beam-forming technique have a different gain in different direction as against omnidirectional antennae that have a uniform gain in all directions. This can cause signal strength received at a receiver to be increased significantly or decreased accordingly by simply by changing the orientation of the directional antenna. In such cases, localization and positioning are major issues.

Directional antenna has larger range of transmission due to higher gain compared to omnidirectional antenna. They are able to receive weak signals. SAMAC is a MAC protocol that employs directional antenna. It is an integrated cross-layer protocol and contains a complete set of communication mechanisms along with being equipped with sectored antennas. Omnidirectional antenna transmit energy in unnecessary directions and thus their transmissions block the communication channel whereas directional communication centres all the radiated power at the intended target thereby reducing the power required for a the respective domain or range. In close proximity multiple communications can be done using directional antennae. Signal interference is a noticeable issue in this case causing energy wastage. Another limitation is that directional antenna may call for adjustments pertinent in event of mobile node or multichannel environment. Deafness problem can be amply serious to counterweight the advantages of beam-forming by directional antennas, if left unaddressed.

vi. Clustering: Clustering offers four-dimensional bandwidth reuse. It simplifies routing decisions ultimately decreasing the energy dissipation of the entire system by reducing the number of nodes participating in long-distance communication and provides scalability as well as robustness to the network. Cluster heads (CH) collect data, aggregate it and then forwards it. Communications can be intra-cluster (between nodes and CH within a cluster) or inter-cluster (between one node in a cluster and another node in another cluster). Different clustering techniques follow various approaches and random schemes to nominate CH and form clusters. Clustering schemes can be categorised into four categories: Hierarchical, Weighted, Heuristic and Grid. Example: HEED was proposed to reduce the number of nodes competing for channel access and better CH nomination algorithm than LEACH. Other examples are DWEHC, CLUBS, MOCA.

1.4.2. Peripheral techniques that support energy efficiency in MAC [37]

These mechanisms are as follows:

1. Routing
2. Topology control
3. Data aggregation

Routing: Multi-hopping as compared to single-hop networks can save a significant amount of energy. Sensor Protocols for Information via Negotiation (SPIN) is one of the most primitive adaptive protocols proposed. SPIN publicizes total information at each node to every other node in the network based on supposition that all nodes in the network are prospective base stations. Routing protocols can be classified to be either data centric, hierarchical or location or based on network flow and QoS awareness.

Topology Control*:* Topology management techniques periodically select nodes to build up an infrastructure that allows other nodes to sleep and power off the radios to preserve energy. An optimized topology maintaining efficient and least cost connectivity is a wide research area. By integrating power control and power management, evident improvements on network can be achieved. High energy efficiency, topology control algorithms and protocols are the goals which can be achieved by: considering power control and sleep scheduling jointly, being aware of traffic load, and implementation done in conjunction with routing. Power control implies minimum transmission power while still preserving the required properties. Sleep scheduling is concerned with saving energy by putting jobless nodes into the sleeping mode.

Data aggregation: Nodes enabled with data aggregation capability, transmit fewer data flows reducing the total energy consumption. The central concept is to aggregate similar packets from multiple nodes to singly transmit it, thereby reducing number of transmissions. In sensor network, data aggregation using a cluster based or a tree based approach. LEACH and PEGASIS employ clustering technique. Hybrid Indirect Transmission or HIT is a hybrid of LEACH and PEGASIS. It consents for multi-hop routes between CHs and other nodes. It is advantageous in a densely clustered network but formation of data aggregation tree that is optimal is generally NP-hard.

1.4.3. Classes of organization of MAC protocol

Three different classes of organization are: random access, slotted access, and frame based access. [36]

Random access: In random access technique, nodes do not compete for accessing the radio channel in an organized way, or time. MAC protocols employing random access techniques transfer the costs from the receiver to the sender by extending the preamble to reduce idle listening enabling nodes to sleep for maximum time and check the channel only periodically.

LPL AND PREAMBLE SAMPLING: Periodic sampling occurring at the physical layer, and MAC protocol of the link layer combined with CSMA is called LPL (Low-Power Listening) and combined with ALOHA it is called Preamble Sampling. Further refinement is called MPS protocol where MPS indicates for Minimal Preamble Sampling. The optimization reduced the need for long preambles in the case of low traffic. Here, sender sends strobed preambles. Example of LPL is BMAC and XMAC and of preamble sampling is WiseMAC, CSMA-MPS. PicoRadio project[36] was the first to propose an innovative technique of using a wake-up radio and detailed the design of passive components that would consume very little energy. The limitation of the technology was that they were quite susceptible to noise (ex. generation of false alerts/alarms), and used broadcast signals (thus awaking all the neighbours). Such extremely small radios are unable to even encode a few bits of any information unit. A better option proposed was to use LPL combined with the wake up radio project as part of Sparse Topology and Energy Management (STEM) protocol. The Rate Estimation MAC (RATE EST) proposed to deal with the problem of waking up all neighbours. Basically wake-up call types of broadcasts can be stopped by piggybacking an explicit interval on a message informing the receiver of next wake up time. If sender is capable of estimating the consecutive point of time when it can forward a message, there is no need of broadcasting wake-up signal. Otherwise, the sender has to either wait until the next scheduled wake-up time of the receiver, or make wake-up call to all neighbours.

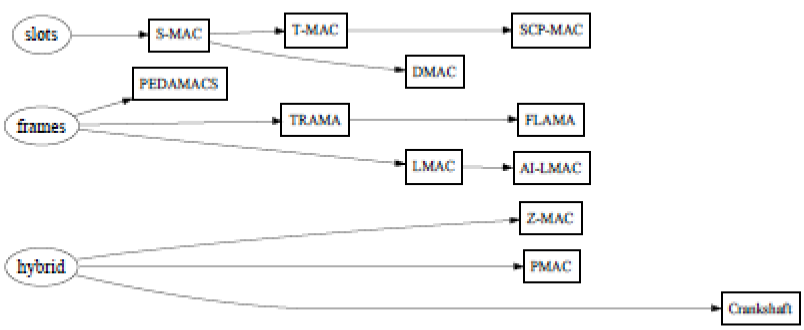

Fig 10: Organization classes are illustrated in diagram[]

Slotted access: Basically this class of protocol is contention-based protocols and the underlying idea is agree nodes on a conjoint sleep/active pattern thus enabling their radios to function at an arbitrarily low duty cycles. Slotted access technique necessitates nodes to synchronize with respect to a time of reference enabling them to wake-up mutually at the start of each slot, exchange memorandums, and then sleep for the remaining slot. Example: S-MAC, TMAC, SCPMAC.

Frame based access: The last class of MAC protocols further groups slots into frames and schedules which node will send in each slot. Frame-based protocols differ from slot based protocols in the manner slots are assigned to nodes. Time is split into frames.
Frames comprises of a fixed number of slots. Example: Classic TDMA, LMAC. LMAC employs a distributed slot-selection mechanism that can self-organize a multi-hop network into a clashless schedule.

1.4.4. Classification based on design approaches

MAC protocols can be allocation based, contention-based and hybrid MAC protocols based on design approach. Allocation or schedule-based protocols are those in which admittance to the channel is served by node's schedule. Channel access in it is restricted to one sensor node at one time. They demonstrate a duty-cycle built-in with the collision-free nature that is inherent too. Both these features ensure low energy consumption, based on the pre allocation of individual resource to the individual nodes. TDMA, FDMA, CDMA are representative allocation based techniques. ALOHA and CSMA are representative contention based protocols. There are two varieties of ALOHA, pure and slotted. In pure ALOHA a node transmits a packet once it is generated or in the next available slot in slotted ALOHA case. Contention-based duty cycle MAC do not serve pre-allocation of resources to one node. Here, there is a common radio channel for all the nodes and are allocated on demand. They are categorized into synchronous and asynchronous MAC protocols. Synchronous approaches such as SMAC, TMAC etc. synchronize with neighboring nodes. Asynchronous protocols such as BMAC, XMAC, WiseMAC etc. permit nodes to operate independently. Each node is on its own duty-cycle schedule. They employ LPL technique. In LPL, before transmitting data, a sender transmits a preamble having life equal to the sleep period of the receiver. Sender-initiated MAC protocols are those in which communication is initiated by the sender by sending preamble or strobed preamble (eg. WiseMAC). In receiver-initiated MAC, receiver initiates the data transmission as in RIMAC for example. Hybrid MAC protocols converge contention and allocation access scheme. In this class of MAC protocols, characteristics of both are shared. They are the most efficient MAC protocols of all three. For example: ZMAC, PMAC etc.

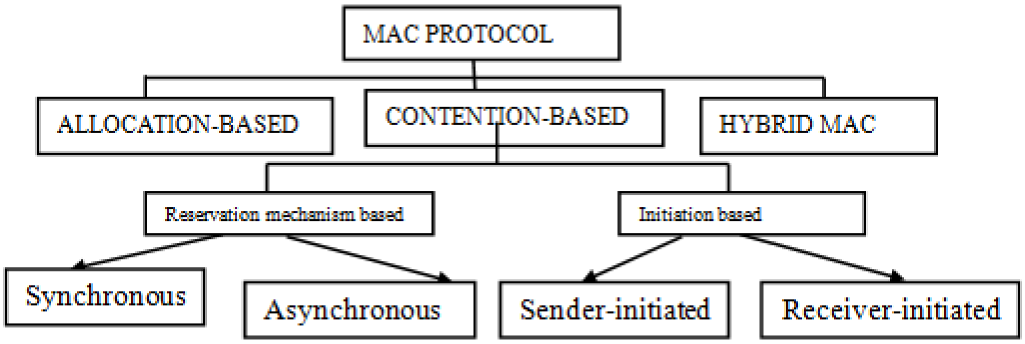

Figure 11: MAC Protocol classification

IEEE 802.11 MAC is the first WLAN to be proposed with a virtual carrier sense mechanism with DCF using CSMA/CA mechanism for medium access.

Chapter 2
LITERATURE REVIEW

Receiver initiated MAC protocol for enhancements for multi-hop WSN, Balvinder (2004). By making MAC receiver-initiated, interferences caused due to presence of hidden nodes was controlled. However it could not be eliminated completely. The receiver-initiated approach imposes requirement on every node. This brings side-effects on performance. BEB (Binary Exponential Back-off) scheme and can at times aggravate fairness problem. Quality of Service (QoS) assurances are hard to provide, implement and maintain without being unbiased. Also the approach has limitations on parallel transmission due to exposed nodes problem. [11]

Pattern MAC, by Tao Zheng (2005): Nodes in this protocol gets traffic information through pattern generation and exchange. These patterns are bits string indications over numerous slot times, which in combination with patterns of node's neighbor, are used by nodes for determining its tentative schedule. Over a slot time, 0 and 1 in string indicate sleep and wake up intention of node respectively. Time is divided into STF (Super Time Frame), which is combo of two sub-frames: PRTF (Pattern Repeat Time Frame) and PETF (Pattern Exchange Time Frame). Patterns are repeated by nodes in PRTF and exchanged during PRTF. PMAC possesses a great and varied scope [29].

MAC layer energy concerns in WSN: Solved by Bram Kersten (2006) studied WUR-MAC (Wake-Up Radio MAC): The concept of wake up radio introduced the idea of equipping sensors with a low power radio majorly concerned with detecting of any incoming calls, transmitting remote interrupts or waking up the main radio in scenes of necessity. Also nodes also need to be provided with a low power passive circuit for detecting, filtering and amplification of signals from wake up radio. Mechanisms used are Off-On Keying and three step process: dumb power detection, address decoding and main radio. A lot of hardware transformations, constraints and quantitative parameters make the implementation of the wake up radio prototype a laborious task but the analytical model of latency and energy consumption proves

WUR-MAC efficiency. It basically deals with a hardware prototype that will meet the functionalities underlined above. MAC protocol for WUR does not need much of synchronization. For example, MAC protocol that combines CSMA and techniques of spread spectrum, for channel negotiation and collision reduction and retransmission by implementing multiple channels respectively. For WUR-MAC, average number of hops is reported to be larger because its range is short. Extra power consumption is function of wake-up signal mainly since main radio is powered up only for data transmissions, but this is very low. Directional antennas, range of wake up signals, an alternative for OOK, multi-stage amplification are some scope of developing an ideal wake up radio/WURMAC. [13]

Receiver initiated directional MAC, Masanori Takata (2006): Directional MAC protocols that used directional antennas popped up various problems along with improvements. One such problem is deafness in which the intended receiver with its beam pointed away from transmitter’s direction caused transmitter to repeatedly attempt contacting. RI-DMAC overcame this problem by having both sender initiated transmission as default scene and receiver initiated transmission mode was initiated in case deafness occurred.[21]

iQueue-MAC. Shuguo Zhuo (2007): In this MAC a variable TDMA period and a CSMA period are incorporated to deal with adaptive traffic. It allocates vTDMA (variable TDMA) period instead of prolonging CP. Router allocates slots to senders who have requested in vTDMA period. The zist of the whole mechanism is that the MAC dispatches packet transmissions in TDMA phase as soon as queuing starts to build up. The author implements the iQueue MAC on IEEE 802.15.4 supporting STW32W108 chip. [30]

FAMA/TDMA hybrid MAC for WSN, by Nuwan Gajaweera (2008): A TDMA protocol based on demand assignment combined with Floor Acquisition Multiple Access protocol, a new hybrid protocol is proposed. Sensor sense the medium and buffer the data. When the node has entered the domain of BST (base station), it will have maximum data to transmit. Such MAC are effective at tracking migration or flow pattern of animal and water body and the most important application where it can be used very effectively is uploading huge data. Advantages of the MAC includes maximum channel utilization, abolition of an uplink map, occasional contention period advisements. The author simulated the protocol on Avrora [28]. Avrora is a cycle accurate simulator of the Atmel AVR controller. It can also simulate MICA2 motes.

YMAC, Youngmin Kim (2008): YMAC is TDMA and LPL based multi-channel protocol. Under easy traffic, it achieves low duty cycle while it effectively transmits messages in bursts in heavy traffic conditions. Each fixed length frame in time consists of a unicast and a short broadcast period. Nodes sample the medium only in its own unicast receiving time slot. A lightweight channel hop technique and an algorithm for generating unbiased sequence number are proposed in the paper to deal with packet queue or falling to lengthy wait of unicast messages. Contention winner can only broadcast data in broadcast period. Control messages (CM) are periodically broadcasted by one or several sink nodes. CM is responsible for initiating and maintaining network and also synchronizing time and compensating time errors caused due to clock drifts. If CM is not received by a node for the predefined wait time, which is thrice of CM, node partitions from the network. [16] Assignment of time slot can be coordinated or distributed. Medium access in YMAC is synchronous and LPL based. Finally YMAC is implemented in RETOS on TmoteSky motes and simulated in both single and multi-hop scenarios under consideration of time sync errors.

Receiver-Initiated MAC, Yanjun Sun (2008): Some basic existing works related to receiver-initiated transmission are LPP (Low Power Probing) in Koala Systems and PTIP (Periodic Terminal Initiated Polling) for infrastructure WSN. This protocol was implemented in TinyOS in a test-bed of MICAz mote. In RIMAC, if receiver R senses medium to be idle, base beacon with no BW field is broadcasted otherwise it backs-off and transmission of the beacon is done in later attempts. Sender node S starts its data transmission immediately on receiving beacon. R acknowledges the receipt of the data frame by sending another beacon called ACK beacon whose Dst. field is set to S. This beacon is also the invitation for a new data frame transmission. Other nodes except S overlook this Dst. Field, thereby counting it as a nod for different data transmission. The node is active for spare time called dwell time, after it has successfully received a data frame. In this dwell time, queued packets are sent to the node immediately. In RI-MAC, beacon frames are the protocol driving entity. A beacon possessing only "Source field" is base beacon. Size of a beacon is saved in frame length field. Dst. address and BW field for back-off window size are two elective fields. Only beacon and data transfer occupy medium thus increasing the capacity of the network. Medium access is responsibility of receiver making RIMAC highly efficient. Sender's and receiver's duty cycle schedules are still kept decoupled. For dealing with collision detection and retransmissions, the maximum delay before the data frame's arrival, is used by receiver. The size of back-off window is explicitly under control of the receiver. The receiver detects the SFD (start of frame delimiter) to learn the details of an approaching frame. If some channel activity is detected by the CCA check in absence of SFD, the receiver concludes of a collision and generates a new beacon with the large BW value. In situations when receiver node for some sender is already active beacon-on-request optimization is used for this sender, when it wakes up. [10]

Protocol design and implementation for wireless sensor network , a paper by Piergiuseppe Di Marco (2008): The cross layer MAC is an innovative approach to design MAC protocol by jointly optimizing and designing networking layers for control and information exchange. It exploits protocol layers interaction to improve performance. A major concern in such design approaches is the parameters to be shared by the layers and their robustness to dynamic network behaviour. [2] Some major examples include SERAN, Breath protocol, LEACH, SPEED, HEED. **SPEED** uses SNGF (Stateless Geographic Non-deterministic Forwarding) routing mechanism. It is a stateless, localized algorithm with minimum overhead for controlling. It also uses MAC and network layer adaptation combo to improve the end to end delay and provides a good response to congestion and voids. **SERAN:** It uses hybrid TDMA/CSMA MAC protocol to combine randomized and deterministic components to generate a two layered clustered protocol. TDMA is operational at the level of cluster, thus reducing rate of wake up while CSMA provides for robustness and acknowledgement based contention scheme to deal with unreliability and duplication respectively. The author studies the SERAN protocol in a great detail in his paper**.** SERAN uses a token passing procedure to ensure nodes-cluster synchronization; to allow initialization and self-configuration; and to allow amassing of new nodes. A token is a precise message carrying the information on a TDMA-slot and a TDMA-cycle, the schedule (transmitting and receiving) of a

TDMA-cycle, a synchronization message that informs the current execution state of the TDMA-cycle. A network initialization algorithm, token refreshing procedure are some other important schemes used in SERAN. The author demonstrated how to implement MAC protocol in a test-bed environment in WSN. From here on, the matter deals with protocol implementation concepts. There are various hardware and software technologies required for the same. Here we review Tmote Sky platform and TinyOS. OS for WSN nodes are typically less complex and must fulfil the listed requirement: robustness, low resource usage, multiple service implementation, adaptability to evolutions, adaptability to application requirements. Most of the currently used OS for sensor nodes let the programmer be the judge of timing the on-off schedule of the sensor (manual management). But future needs OS that can automate the on-off schedule according to sensor, environment and application specific requirements thereby eliminating risk, mishaps and mistakes of manual handling and improving the usefulness of the duty-cycling mechanism. TinyOS is an embedded OS specially designed for use in WSNs. In TinyOS, application is compiled and used for programming a single node. It uses “Hurry Up and Sleep Philosophy”; according to which when a node activates itself for an event, it must execute the associated action immediately, then go back to sleep. The TinyOS 2.x family is a established branch of OS. The TinyOS system, application, libraries are all written in nesC, a version of C designed for programming embedded systems.

The features of TinyOS 2.x are listed below:

1. Resource constrained concurrency: Whenever shared variable is accessed, multiple stacks are kept in memory and each thread can potentially interact with any other. TinyOS offers different levels of concurrency, in a structured event-driven execution.
2. Structured event driven execution: The interface of primitive components encapsulating radio, timer or any other hardware, reveals the hardware operations and interrupts; the state and concurrency is that of the physical device. Higher-level components encapsulate software functionality, but with a similar abstraction. They provide for commands, signal events, and have internal handlers, task threads, and state variables. It is particularly achieved by component replacement.

3. Components and bidirectional interfaces: TinyOS support component composition, system-wide analysis, and network data types.
4. Split-phase operations: Since TinyOS is non-preemptive and does not support blocking operations therefore, all long latency operations need to be accomplished in a split-phase fashion. In split phase operation, the operation request and the completion signalling are separated in bidirectional interfaces.
5. Communication and networking
6. Sensing: Sensor drivers scaling from low-rate, low-power sampling (of environmental factors) to high-rate, low-jitter sampling (of vibration or acceleration).
7. Storage: Non-volatile storage is good for logging, configuring parameters, files, and programming images.

The default MAC code of TinyOS 2.x implements two types of acknowledgements:

1. Hardware acknowledgements: It is directly employed at the transceiver.
2. Software acknowledgements: It is labelled by the interface Packet Acknowledgements.

Tmote Sky was designed at the University of California, Berkeley with the goal of fault tolerance and ease the development. It is a node platform for low power and high data-rate sensor network applications. It is the successor of TelosA and TelosB, the most popular research platforms of that time. The Tmote Sky platform offers vertical integration between the hardware and the TinyOS. The module of Tmote Sky has sensors, radio, antenna, microcontroller cohesive with programming capabilities. It uses low power TI MSP430 microcontroller for low power operation. It has Chipcon CC2420 radio. This radio can be easily configured for many applications with the default radio setting providing IEEE 802.15.4 compliance. The radio is controlled by the microcontroller through the SPI port and provides fast data rate and robust signal. It can be shut off for low power duty cycled operation. Tmote Sky has an internal antenna, which is basically an inverted-F microstrip design, with a pseudo omnidirectional pattern that may attain 50 meter range indoors and up to 125 meter range outdoors. The protocol implementation suffers from two major down sides: collision and duplicate packets. Duplication occurs when a packet though correctly received but ACK fails to arrive at the sender may be due to bad channel. Hence, the sender retransmits the packet. It can also happen if in the receiving cluster a node fails to hear the ACK transmitted by some other node thereby causing more than one node forwards the same packet.

BOX-MAC, David Moss (2009): Box-MAC-1 incorporates information at link layer to physical layer sampling, and is suitable for light traffic networks that are highly volatile. It continuously packetizes wake up transmissions on packet radio. The effect is that nodes remain awake only for packets destined for them. Box-MAC-2 incorporates information at physical layer with packetizing at link layer and is effective for heavy traffic networks with low volatility. This helps in detecting if any energy is present in the medium and it decouples size of receive check and wake up packet. Both these MACs have been elaborately compared as an improvement upon and solution to overhearing problem in BMAC and XMAC respectively by implementing them on CC2420 under TinyOS 2.x. Dynamic Box-MAC and its variation built in combination with synchronous MAC protocols are among the various scopes briefed in the paper. [18]

Energy management in WSN by Cseare Alippi (2009) studied challenges in implementing LEACH which are numerous though for instance, it is not apt for deployment in wide coverage area of networks because nodes in LEACH must support different MAC protocols. It also assumes that all nodes begin with same amount of power which is basically impractical. The next assumption that kills LEACH is that nodes near to each other possess correlated data and nodes always shave data to be sent. Dynamic clustering in LEACH also adds extra overhead. In spite of all these constraints, LEACH protocol is a seamless example of a perfect protocol as it uses clustering, data aggregation, and in-network processing in order to provide an optimum performance level. [2]

Improvement of receiver-initiated MAC protocols for WSN by applying schedule by Cristina Cano (2010) : The paper mentions the two drawbacks of RIMAC that the nodes have to broadcast beacon everytime they waking, even though there may be no data for it. The next problem is inefficient broadcasting. The author proposes RIMAC with scheduling to combat this problem.

A technique called distributed learning is proposed which will establish the order of beacon transmissions based on the condition that per wake-up period, only one neighbouring node is allowed to transmit the beacon. [23]

Energy saving in MAC layer: a survey by A.Roy (2010): Basic contention MAC like SMAC (Sensor MAC), TMAC (Timeout MAC) and LPL (Low Power Listening) based: BMAC (Berkeley MAC), WISEMAC, XMAC, CMAC (Convergent MAC), SCPMAC (Schedule Channel Polling MAC) is jotted below. SMAC is low power fixed duty cycle, slotted medium contention; RTS-CTS (Request-to-send – Clear-to-send) mechanism based asynchronous MAC. It suffered from lot of energy wastage, less efficient, high cost and overhead and was thus improved upon to give TMAC. TMAC was effective due to its timeout strategy but suffered from early sleeping problem. In TMAC, events triggered active period of nodes and if nodes sensed nothing on the medium, they timed out and slept. In LPL mechanism, senders broadcast preamble before transmitting data. LPL based MAC reduced power consumption and were based on random medium contention. Nodes in TMAC transmitted messages in bursts of variable length. BMAC uses CSMA (Carrier Sense Multiple Access) technique together with preamble sampling but suffers from problem of overhearing, long preamble and higher latency, cost and contention. In WISE-MAC, preamble is sampled beforehand and nodes sample the medium with independent relative offsets of a common schedule. It suffers from hidden terminal problem, decentralization of sleep and active periods. In XMAC, short strobed preamble was used and it was an improvement over BMAC and WISE-MAC. Also sleeping schedules of sender and receiver are decoupled. CMAC utilizes antagonistic RTS, as well as anycast, and convergent packet forwarding mechanism. It checks channel twice. SCPMAC combines preamble sampling and scheduling. Other mechanisms used by it are adaptive channel sampling, two-stage contention, and multi-hop flooding. [8]

Double-loop receiver-initiated MAC by Hao Liang (2011): The paper proposes a DRMAC to deal with intermittent network connectivity. A contention group CG is a group of local nodes participating in the wireless channel contention for packet transmission to the wandering node. The proposed scheme comprises of two MAC loops: outer and inner. The former performs at a low frequency.

Based on average transmission rate, CG membership is determined by outer loop for spatial diversity, whereas later is performed at a high frequency for the purpose of selecting the transmitter for temporal diversity based on the instantaneous transmission rate.

Adaptive coordinated MAC, Jing Ai (2011): Designed on the grounds of SMAC protocol, major characteristics of AC-MAC protocol include adaptive and dynamically coordinated duty cycle, virtual clustering and synchronization. Nodes contend for the medium when they are awake, the one winning the race takes up transmission while rest sleep. While awake, nodes maintain synchronization and avoid collision in Sync and RTS/CTS interval respectively. The remarkable protocol driving feature is that on the basis of queued packet frequency at MAC layer, it decides the to be traffic density which further is helpful in deciding the number of communication chances to be provided within a cycle. Simulation study of AC-MAC protocol by the author [24] is done on similar terms as SMAC.

QUEEN-MAC, Gholam Hossein Ekbatanifard (2012): A quorum based energy efficient MAC protocol for WSN. MAC protocols that work on similar grounds are QMAC (Quorum-based MAC), TMCP (Tree-based Multichannel MAC), EM-MAC (Efficient Multichannel MAC), IEEE 802.15.4, MMSN (Multi-frequency Media Access Protocol). Quorums can be designed based on tree, grid, torus, extended torus and majority. Queen-MAC is a multiple channel, dyadic grid based quorum system used to provide adaptive low duty cycle protocol for various data gathering and broadcasting purposes. Dygrid lowers network sensibility by providing rendezvous points. Nodes are active in quorum slots and grouped based on their distance from sink. This group number further is used for channel assignment. It sleeps in non-quorum slots, which is independent of active slot. It is energy efficient and provides better network latency. The author simulates Queen-MAC in OPNET Modeler 14.0 [19].

Reinforcement Learning based MAC was proposed by Zhenzhen Liu (2012): Introducing the drawbacks in PMAC [29], the author [32] mentions how computation complexity and assuming that requirement of use of already anticipated channel is known beforehand is quite limiting the protocol for practical implementation.

Reinforcement learning based context is used by nodes in RL-MAC to infer the state of other nodes. After reservation of slots dynamically by RL agent, reward function is formulated that helps the agent learn the optimum MAC policy. The protocol avoids early sleeping, and employs q-learning algorithm.

A Survey on SCHEDULE BASED MAC PROTOCOL was done by Abul Kalam Azad (2013): Some of the major challenges faced in such MAC protocols include time synchronization and resynchronization, schedule assignment, maintenance and adaptation. LEACH (Low Energy Adaptive Clustering Hierarchy), SMACS(Self-organizing MAC for Sensor Networks), TRAMA(Traffic-Adaptive Medium Access), EMACS (EYES MAC), LMAC(Lightweight Medium Access Control), AI-LMAC(Adaptive, Information-Centric and Lightweight MAC), MC-LMAC(Multi-Channel MAC), DMAC (Data Gathering MAC), DE-MAC(Distributed Energy-aware MAC) were compared and discussed along with their advantages and disadvantages. TDMA (Time Division Multiple Access) based protocols among these are LEACH, SMACS, EMACS, LMAC, AI-LMAC and DE-MAC. In LEACH, dynamic clusters of nodes under the control of a cluster-head node are formed. Cluster head node is responsible for creating, distributing and maintaining TDMA schedules with other member nodes of the cluster and therefore it is highly energy consuming perishing out soon. A new cluster-head is selected by non-cluster-head nodes in each round based on the received signal strength of individual nodes. Each round has a setup phase for cluster-head node election and a steady-state or advertisement phase for advertising the nominee nodes. SMACS is based on various assumptions and is used for various scenarios of neighbor discovery combined with link setup, in addition to TDMA slot assignment. TRAMA is based on NP (Neighbor Protocol), AEA (Adaptive Election Algorithm), SEP (Schedule Exchange Protocol). Nodes in TRAMA create schedules on demand, to access a single channel and each cycle of time is made of random and schedule access periods. In EMACS, nodes transmit in their respective timeslots, they do not contend. Sections in each timeslot include: communication request for requesting for data (if any), traffic control for broadcasting schedule table and data for actual uplink/downlink transmission. Nodes in EMAC are on/in standby mode for the entire slot assigned to it, even if it is not functional due to any reason; thus increasing latency. Sleep time in LMAC protocol is adaptive to traffic with reduced frequency of transceiver switches. LMAC timeslots are made of control message and data message period. During control message period of timeslot, network set up is done. AI-LMAC is a variant of LMAC whose essence is parent-child association with gateway and here nodes maintain DDT (Data Distribution Table) to respond to queries. It ensures two-dimensional fairness. In MC-LMAC, bandwidth utilization is maximum due to coordinated transmission over multiple channels, and timeslots also include common frequency (CF) periods and slots in it, in addition to components of timeslot of LMAC. CF is reserved for channel number during which the sender will address destination. An important characteristic of this MAC is that timeslot can be selected from multi-channel as well as over different channel between two-hop neighbour by using localized scheduling algorithm. DMAC enables uninterrupted forwarding of data packets using gathering and prediction mechanisms. An MTS (More to send) packet is sent by sender to its parent, if it is left with packets to send. DEMAC treats nodes in distributed manner i.e. based on its criticality or energy status. Nodes have two timeslots (transmitting and listening) and two phases: normal and voting phase for readjustment of slots, and maintains a receiver table. [7]

EVAM-MAC, Zaher Merhi (2013): The paper proposes an event based MAC protocol with multi hop support for WSN. When an event has occurred, a virtual cluster is created. This cluster contains only the nodes that detected the event. Further, schedule based on TDMA that arranges the data transfer is created for the cluster. This schedule is prioritized based on received signal strength of node. It does not use global slot generation and synchronization thus reducing extra overhead, but still maintains dynamicity. It is particularly employed before nodes have been deployed to prevent a sudden increase of contention. EVAM MAC shifts the contention to control phase. Other techniques used in the rounds of the design approach include neighbour discovery, leader node selection, time synchronization, multi hop resolution. The author has also simulated the results in ns2 and compared it to SMAC. [27]

Packet Duration Value Based MAC, Abayomi M.Ajofoyinbo (2013): According to the idea proposed by PDV MAC, data packets contain duration value (DV) stored in a variable called Network Allocation Vector (NAV) located on sensor node. DV decrements by one each time a transmission of packet is done and NAV timer attached to it fires, when NAV becomes zero i.e. when medium is idle. DV in NAV keeps track of schedules of its neighbouring nodes and thus the time to transmit the pending data frames. After RTS and CTS exchange, receiver receives DV of sender. It then generates several randomized values corresponding to each of its neighbouring nodes in the range 1-100ms. These values are then added to the DV from sender to generate current transmission notification and ultimately broadcasted to all the neighbours of the receiver. The process yields an ordered sleep/wakeup schedule and transmit time of the nodes, in an attempt to prevent collision and idle listening. The efficiency of PDV MAC is tested by simulating it in Visual C# and MATLAB.

Hybrid MAC protocols for WSN , Sumita nagah (2013): The paper gives a brief overview of ZMAC (Zebra MAC), IHMAC (Intelligent Hybrid MAC), HyMAC (Hybrid MAC), PHMAC (Parallel and Hierarchical MAC), ER-MAC (Emergency Response MAC). ZMAC and ER-MAC employ combo of TDMA and CSMA techniques. But ERMAC is advantageous over ZMAC as in it nodes are free to be in or leave the network. IHMAC along with TDMA and CSMA, switches dynamically between link and broadcast scheduling, uses RTS-CTS handshake to perform under wide range of traffic loads. HyMAC combines TDMA and FDMA (Frequency Division Multiple Access) technique thus outperforming protocols like BMAC. PHMAC allocates, synchronizes and does time division multiple access.

Priority MAC was proposed by Sana Ullah (2014): PMAC is based on the novel idea of priority wise resource allocation and CSMA/CA (CSMA/Collision Avoidance). The super-frame of PMAC features one contention free period (CFP) amid two contention access points (CAP1 and CAP2) and beacons at the ends. Beacons are concerned with availability and allocation of resources to the authenticated nodes. A coordinator in CAP accepts requests for TDMA slots, which are actually allocated to CFP of subsequent super-frame. PMAC carries out prioritized CSMA/CA process in the CAP and uses a pre-shared master key which further generates session specific pairwise temporal key after association, for authentication purposes. CAP2 can be optional with functions being handling of pending or incomplete transmissions. Node priority and contention window size are inversely proportional to each other and back-off counter (BC) can be any integer between 1 and the contention window size. BC decrements for every idle slot and reaches zero to initiate the transmission process. The paper reaches for analytical approximations for performance metrics for the protocol and especially for CAP like probability of packet loss, count of corrupted contention free period etc. [15]

Adaptive MAC, D.Antony (2014): Adaptive MAC generates and updates its wake up schedule based on its own energy consumption rate. For instance, node with higher battery power possesses wakes up more frequently compared to its counterparts. Subsequently, nodes undergo self-organization and function phase. Self-organization implies establishing, broadcasting, listening, adopting sleep/wake-up schedule of self and neighboring nodes. In function phase, communication of probes and auto-acks by receiver and sender respectively, over multiple channel takes place. The author does ns2 simulation [26] of the protocol to conclude for its better performance compared to RIMAC.

Advertisement MAC, Surjya Ray (2014): Major features are advertising for contention race and multicasting at the MAC level. In addition to the variable active and sleep phase, ADVMAC has a fixed length sync and advertisement period before active phase, total frame time being fixed. ADV period/time has multiple ADV slots. Random ADV slots are assigned to needy nodes in which it will transmit an ADV packet containing ID of the destination or receiver node. In active phase, nodes that have advertised and received ADV will stay awake to contend and receive respectively. SMAC compromises with maximum latency and minimum throughput and TMAC suffers from early sleeping problem as well as both broadcast packets directly without sufficing to importance of type specification or compatibility leading to energy loss in instances of overhearing. ADVMAC performs better and saves energy by combating overhearing and idle listening in such instances and supports for single hop multicasting. There is scope of introducing TDMA into the concept of ADV-MAC in order to remove the need of RTS-CTS cropping up during data exchange. [9]

Energy efficient routing and fault tolerant clustering using EEPW-MAC protocol, Yasmeen Sultana (2014): This paper proposed PW-MAC (Predictive Wake-up MAC). Though being a receiver initiated protocol, PWMAC is known to reduce duty cycle at sender as well. Sender nodes predict wake up time of intended receiver based on the pseudo-random wake up schedule generator algorithm. Nodes learn each other pseudo-random generators and with clock drift considerations, compute exact wake up time of each of its neighbors or receivers. If not then, when sender has received beacon, it transmits data packet whose header is set with a flag requesting receiver's prediction states. Receiver replies with requested details embedded in acknowledgement message. Thus, compared to RIMAC where sender had to remain awake for the beacon, PW-MAC has reduced this idle-listening time of the sender. For enhanced performance, errors due to difference in actual and predicted wake up time are dealt with on-demand prediction error control mechanism wherein sender holds the responsibility of requesting prediction-state update. PWMAC is evaluated via simulation on ns-2.34 [25] and presents the scope of inculcating geographical routing with it.

Energy usage reduction in wireless sensor network using asynchronous duty cycle MAC protocol by T.Saranya Lakshmi (2014): This paper proposed **WTA-MAC** (Wake up Time Aware MAC) as an improvement over RIMAC and simulated in ns2 environment [12]. The protocol calculates one and subsequent wake up times of the receiver beforehand; based on the arrival rate of data at the receiver. Sender keeps this schedule and activates immediately before the wake up time of receiver. Receiver waits until timeout to receive the packet. WTA-MAC reduces idle-listening time, overhearing and collision.

GAME-THEORY BASED HYBRID MAC, P.Raja (2014): GH-MAC is combination of G-ETDMA (Game based energy efficient TDMA) that deals with intra-cluster communication between cluster members with head nodes and G-nanoMAC (Game theory based nanoMAC) , a CSMA/CA based MAC protocol used for inter-cluster communication between head nodes. Concept of dynamic repeated game is used to evaluate energy utilized by nodes and cluster heads in a cluster. Game is formulated based on how cluster heads strategize their schedule based on traffic load. GHMAC is analyzed by author using MATLAB 10. [20]

Chapter 3
WORK PROPOSED

3.1. OBJECTIVE

The objective of the MAC protocol is regulating admittance to the shared wireless medium such that the performance necessities of all the primary applications are fulfilled. The sensor nodes must consume extremely less battery power, operate in volumetric compactness, should have low production cost and be indispensable, be self-sufficient and operate without supervision, be adaptable to the environment. Keeping eye on these specifications several MAC protocols have been designed, developed and implemented; with hybrid and cross layer MAC being the most successful of all. Hybrid MAC protocols converge contention and allocation access schemes. The resultant class of protocols share the characteristics of both. In this report, a hybrid of TMAC and RIMAC has been proposed which converges the central idea of both the concepts to enhance sender-receiver coordination. It has been named BCAMAC for "Beacon Activation Medium Access Control". The objective of this report is present the feasibility of BCAMAC protocol, and analyze and compare its performance with respect to the base MACs .e. RIMAC, TMAC in terms of three parameters: delay, latency, throughput.

3.2. PROBLEM DEFINITION

Preserving the vital schemes and properties of TMAC's dynamic timeout and RIMAC's receiver-initiation, the novel hybrid MAC enhances the efficiency of the proposed protocol by inculcating beacon reception as an activation event. Every node acting as a receiver R wakes up intermittently, according to its schedule to check if there is any pending data intended for it. If the medium is idle R immediately broadcasts a base beacon announcing that it is awake and ready to receive data frames. A sender node S with pending data to send was active silently. The activation events enumerated in the TMAC protocol will now have an extra event "Reception of a base beacon" included into the list. The length of the active time is determined dynamically.

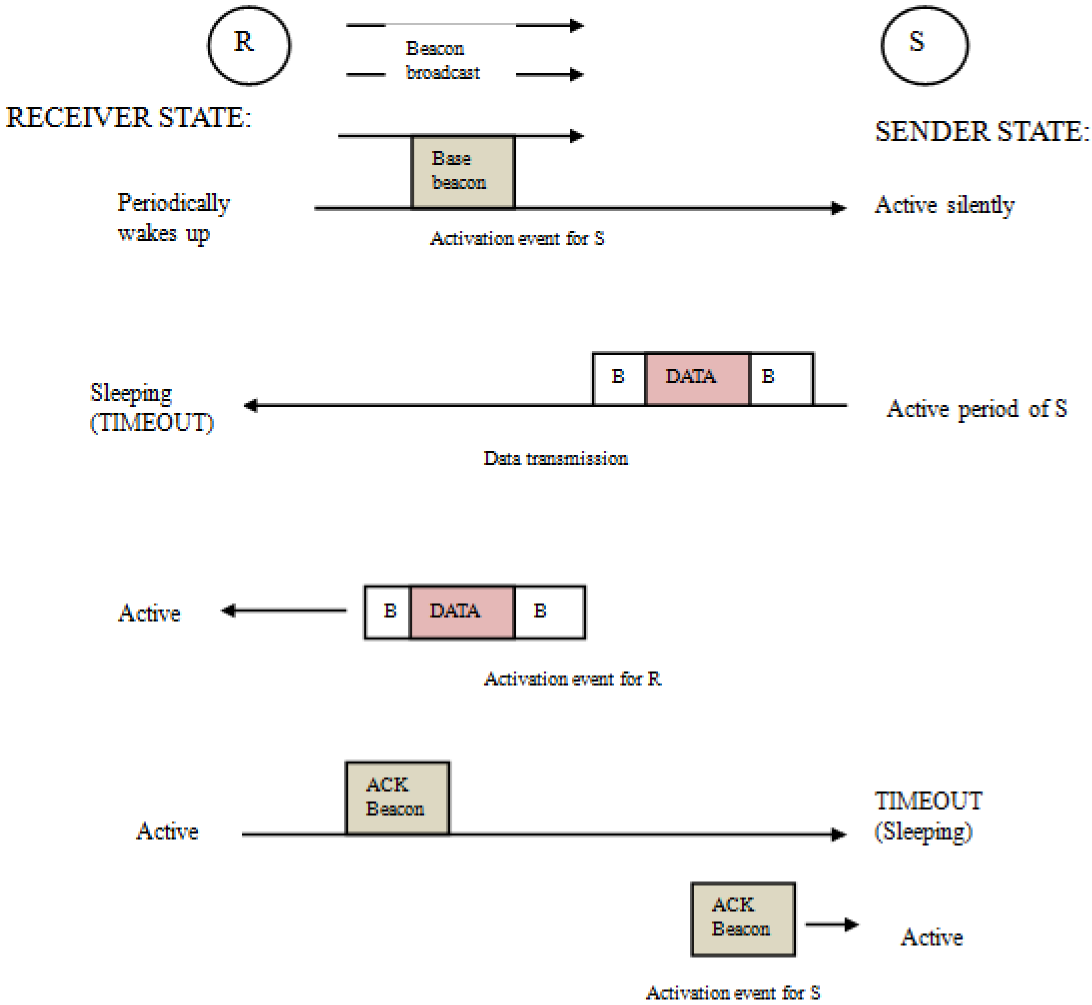

Figure 12: The formulated mechanism for BCAMAC protocol

Upon receiver initiation through base beacon, sender starts communicating to the receiver node. Upon reception of ACK beacons it continues transmitting and receiving packets. The communication will end only when the sender senses no data packet from any other node or any beacons. And, finally it will timeout. If still receiver was left with some pending data frames, it can still send those to the sender as reception of any data on the radio is among one of the activation events of the node, thus bringing it back to its active period. This packet can also be buffered. Above concept will be verified by preliminary simulations. Simulation experiments will further verify the performance, power efficiency and other trade-offs with varying traffics and contending flows of BCAMAC. Simulation experiment will also analyse other performance matrices of the protocol as well solutions dealing with collision and other causes of energy waste.

3.3. METHODOLOGY

Network Simulation is a technique used to model and analyze the behavior, working and characteristics of the existing or researched network scenario, using programming. Various types of simulator tools, scripts, programming languages and platform have been developed for the purpose. Almost all the presently available simulator tools come with built in support for very basic to advanced network fundamentals and protocols. Earliest simulator was REAL. Among the simulator tools available currently are NetSim, NS, OMneT++, PhySim, Traffic, Shunra Virtual Enterprise, Opnet, Cnet, OptSim, GloMoSim, GTNeTS, Parsec, QualNet Developer, NCTuns, Performance PROPHET, and SWANS. Now to analyse and compare the work proposed idea or model or simulation, graph generating tools like tracegraph, xgraph, GnuPLOT.

3.3.1. TOOL USED

A simulator model of a practically used system is essentially a generalization. Network simulator: NS is a discrete packet-level simulator, focusing networking research. It supports for simulation of TCP, routing, and multicast protocols over both wired and wireless networks. A series of discrete event network simulators: ns1, ns2, ns3 have been developed. NS1 was derived from REAL and is no longer in use. It used scripting based on Tcl. Ns3 is the latest version of ns available since 2008. It uses C++ and Python with scripting capability but is not backward compatible i.e. ns2 simulations cannot be simulated on ns3.

NS2 is widely used. Ns2 entails two languages: C++ and OTcl (Object-oriented tool command language). C++ defines the internal mechanism of simulation objects or the core of ns2. The OTcl collects and configures the objects as well as schedule discrete events. OTcl is an extension of Tcl scripting language. C++ and Otcl are associated using TclCL. Variables in OTcl are mapped to C++ objects. C++ is used for “data” and OTcl for controlling actions. Presently ns2 consists of over 300,000 lines of source code including 1,00,000 source code lines of C++, 70,000 lines of OTcl, 50,000 lines of code of test-suite, examples, docs. Platforms supporting ns2 are most UNIX and UNIX-like systems (FreeBSD, Linux, Solaris), Windows 95/98/NT/XP with Cygwin.

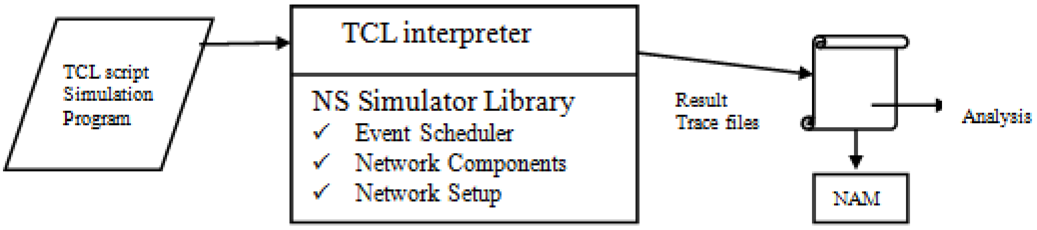

Figure 13: Simplified user's view of ns2

3.3.2. ENVIRONMENT SET-UP

Ns2 provides numerous built-in C++ objects. It is recommended to use them to set up an environment for simulation, using a Tcl simulation script except in case one is an advanced user.

For setting up the environment for simulation, the most commonly used ways on various platforms are briefed below:

1) Cygwin on Windows
2) Linux: Examples are Ubuntu, RedHat, Debian, Fedora. It can also be used on virtual machine.
3) Unix

Building ns:

Ns can be built from pieces or directly installed from open source all-in-one software. When building ns from pieces mostly done when using Unix, the packages required are downloaded separately and then integrated. The packages contained in ns are Tcl, OTcl, TclCL, ns latest version. Other optional packages include xgraph, nam, perl, tcl debug, demalloc, sgb2ns conversion program, tiers2ns conversion program, cweb and sgb source codes.

Steps required for building ns:

1) Update the platform being used using apt commands.
2) Install the essential packages required by ns.
3) Extract from the zip folder.

4) In the terminal, move to the directory where ns has been unzipped.
5) Use the following command to install

   ./install

6) Set the environment variables in bash file.
7) Validate the installation using the following command

   ./validate

3.3.3. SIMULATION

General process of creating simulation

The steps involved in creating a simulation include the following:

1) Defining topology
2) Developing model
3) Configuring nodes and links
4) Executing
5) Performance analysis
6) Graphical visualization

Ns2 modules can be developed in any of the three file formats mentioned below:

1) C++ file: These files have .cc extension
2) Header files: Their extensions are .h
3) Tcl files: Tcl files are most commonly used and come with simple .tcl extensions.

3.3.4. COMPILING FILES

In command prompt, path of the directory containing the file is typed. Subsequently, code to run the TCL file is typed in command prompt as below:

ns <filename>.tcl

Tracefiles store overall network information. They bear .tr file extension. Command "set nf" is the syntax used to create a tracefile. Awk scripts are used to process data from tracefiles or any other text file. For using awk, gawk needs to be installed in the system and it can be done by typing following command in the Linux terminal:

Sudo apt-get install gawk

Nam is an animation tool built based on Tcl/Tk. It is majorly useful in visualizing ns simulations and real world packet trace data. Steps to use nam are: Produce a nam trace file containing topology information like nodes, links, queues, node connectivity etc. They also contain packet trace information. NSG (Network Simulator Scenarios Generator) is a Java based tcl script generator tool. It can be used on any platform.

3.3.5. Coding in ns2

NS2 programming steps:

1.Create the event scheduler.

2.Turn on tracing.

3.Creating network:

  a) Computing setup routing using routing protocols

  b) Creating transport connection using agents

  c) Creating traffic

4. Monitoring :

  a) Visualization using nam.

A sample format for coding in ns2 is shown below.

```
set ns [new Simulator]                    } Creating simulator object

set tracefile [open out.tr w]             } Opening trace file
$ns trace-all $tracefile

set nf [open out.nam w]                   } Opening the nam trace file
$ns namtrace-all $nf

proc finish {}
{
      global ns tracefile nf
      $ns flush-trace
      close $nf
      close $tracefile                    } "finish" procedure
      exec nam out.nam &
      exit 0
}
```

set n0 [$ns node] set n1 [$ns node] $ns simplex-link $n0 $n1 1Mb 10ms DropTail	Creating topology - set n0 nodes…. - $ns duplex-links…
set udp0 [new Agent/UDP] $ns attach-agent $n0 $udp0 set cbr[new Application/Traffic/CBR] $cbr attach-agent $udp0 set null0 [new Agent/Null] $ns attach-agent $n1 $null0 $ns connect $udp0 $null0	Creating agents at transport layer and application layer
$ns at 1.0 "$cbr start" $ns at 3.0 "finish"	Scheduling Events - $ns at 1.0 start and at 3.0 finish
$ns run	start the simulation.

The other codes for the required functionality and traffic flows can be added after nodes have been created and before simulation is ended.

3.3.6. Generating graphs

For generating graphs to analyse ns2 codes, tools that can be used are xgraph, trace graph, gnuplot etc. and in windows environment sigmaplot can be used. For generating graphs for our MAC protocol, tracegraph has been used. This software is not maintained by anyone still it works fine as well as is free.

Chapter 4

RESULTS AND DISCUSSION

In the RIMAC scenario two-way ground propagation model has network interface type is phy/WirelessPhy, IEEE 802.11 base MAC is used, interface queue type is drop-tail and priority queue with maximum capacity of 50 packets. DSDV (Destination-Sequenced Distance Vector routing) routing protocol is used for routing the packets in the simulation.

The following code snippet shows the MAC framework used in the simulation.

```
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
#set val(mac)   Mac/RMAC                   ;# MAC type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     $val(node_num)             ;# number of mobilenodes
set val(rp)     DSDV                       ;# routing protocol
```

The network animation of the proposed idea is shown below in the snapshot. Topology of seven nodes can be seen from the snapshots. The first snapshot depicts the transmission of beacons. Sender nodes are activated when they have received the base beacon and they start data transmission as depicted in the next snapshot. Consecutive snapshots depict how link failure cannot stop transmission as other active links take over. And finally the timeout takes place.

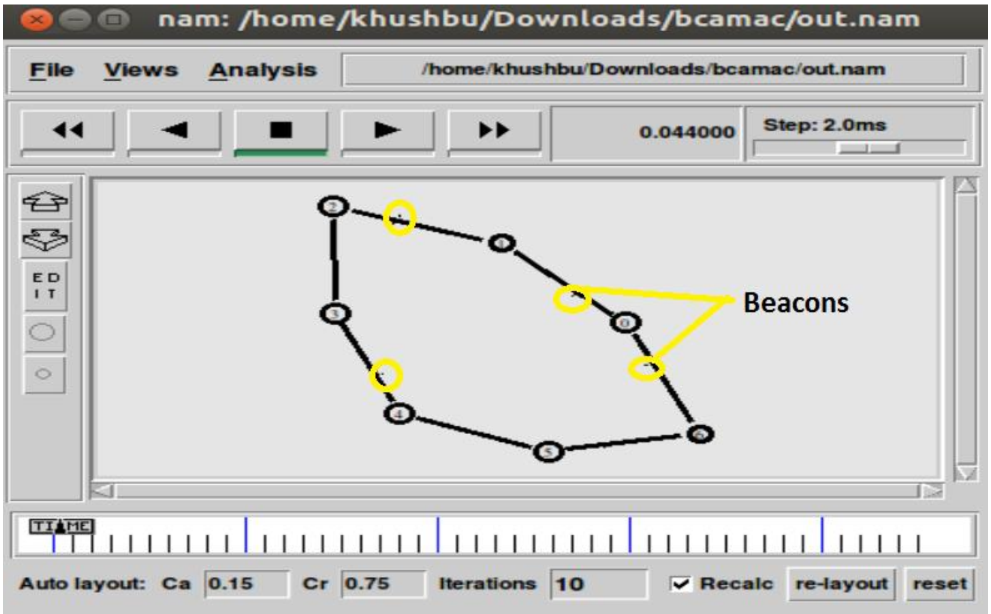

Figure 14: Snapshot: Transmission of beacons

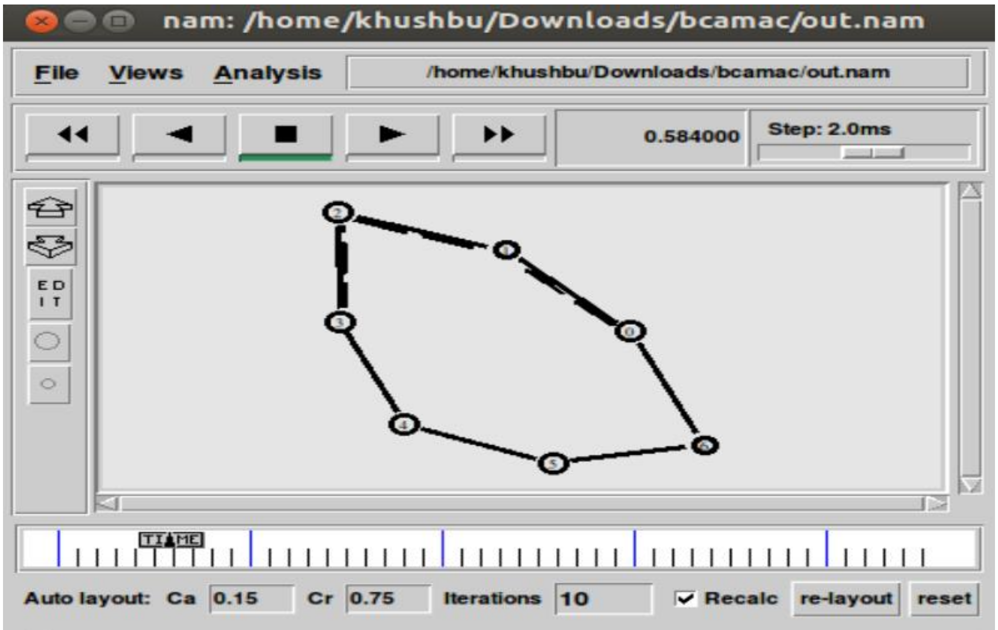

Figure 15: Snapshot: Initiation of communication

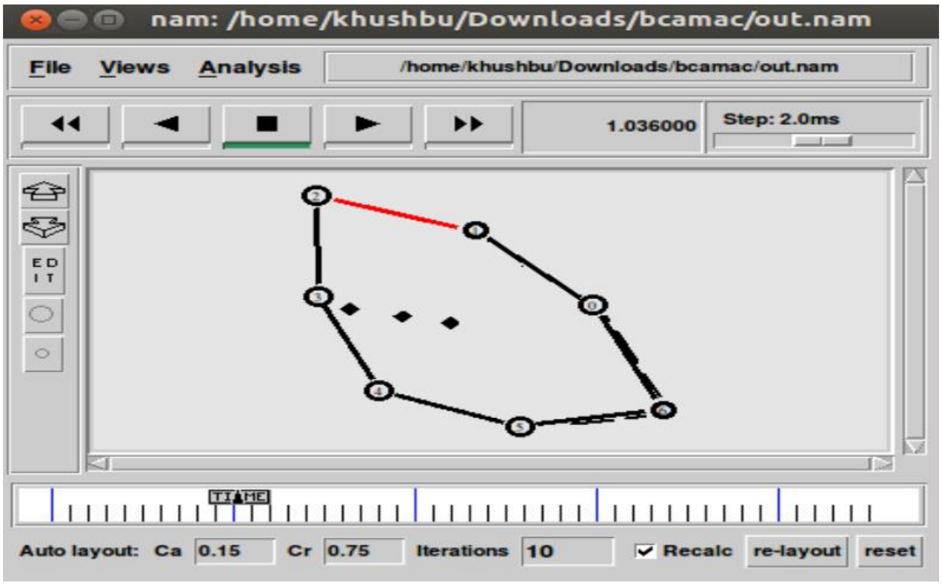

Figure 16: Snapshot: Link failure, Loss of packet, Resuming transmission from another link

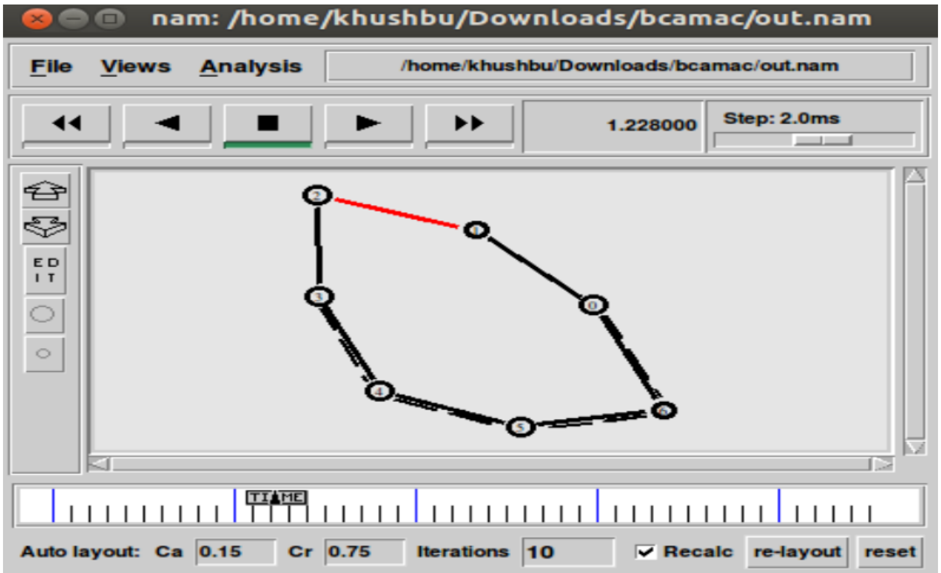

Figure 17: Snapshot: Normal communication even after link failure

Plotting the following parameter analysis of the BCAMAC protocol in the tracegraph.

1. Throughput vs end-to-end delay
2. Packet size vs minimal throughput of generating packets at the node
3. Packet size vs end-to-end delay

From the graph in figure 14 below it can been seen that with increasing throughput , average end to end delay increases until a point after which the graph starts to attain almost normal behavior.

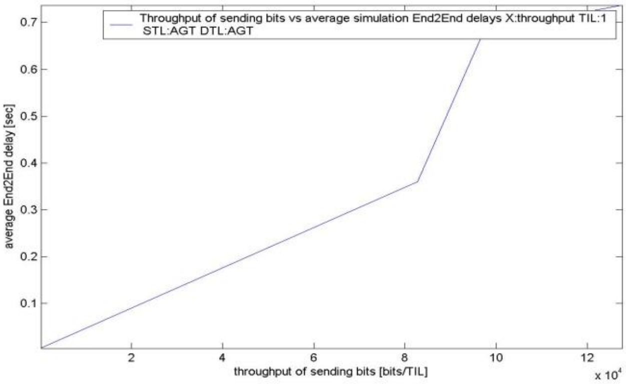

Figure 18: throughput vs. average end-to-end delay

After initial transmission phase, when packet size increases minimal throughput attains a maximum value alarmingly but soon decreases to become zero and finally attains a intermittent increase and decrease pattern as evident from figure 15.

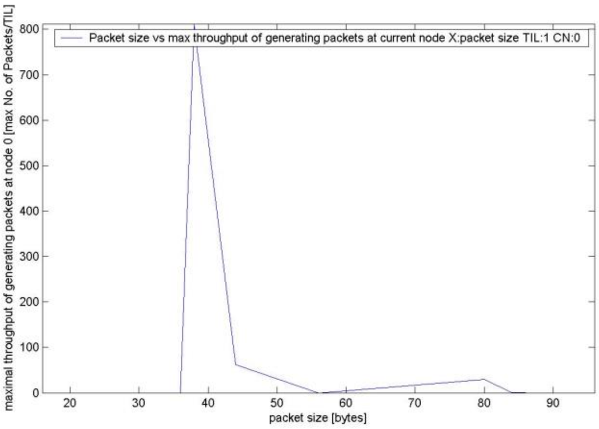

Figure 19: Packet size vs. minimal throughput

With increasing packet size, throughput may not be vulnerable but end-to-end delay decreases alarmingly to zero for the proposed MAC thereby declaring its packet delivering efficiency. This observation is shown in figure 16.

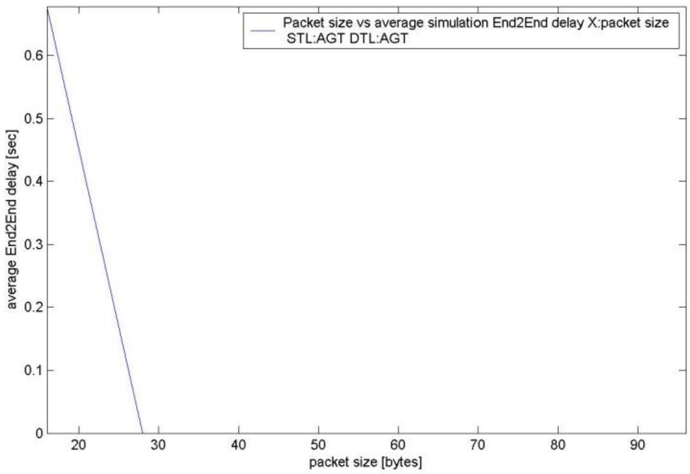

Figure 20: Packet size vs. end-to-end delay

Three dimensional graphs of communication parameters taking sending node functional at Y axis and receiving node functional at X axis and time on Z axis perpendicular to plane. The illustration is used to convey how significantly end-to-delay has been reduced in the proposed MAC protocol.

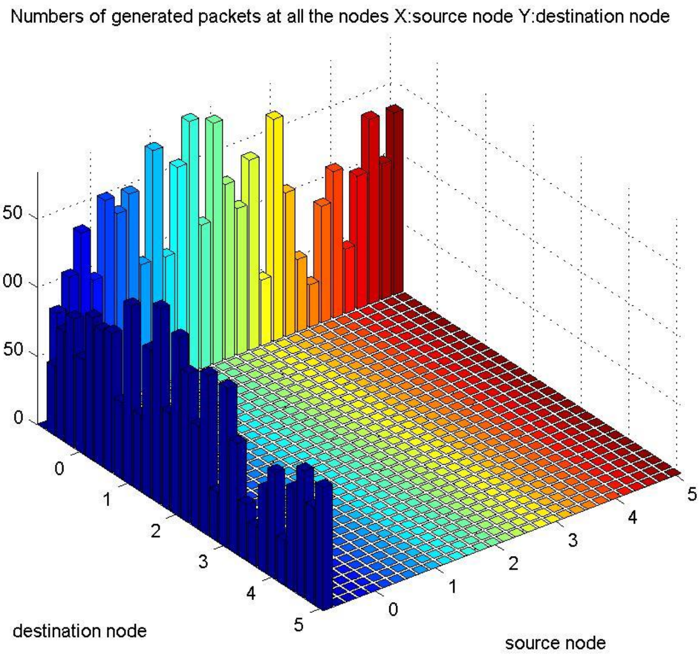

Figure 21: Parameter: Number of packets generated

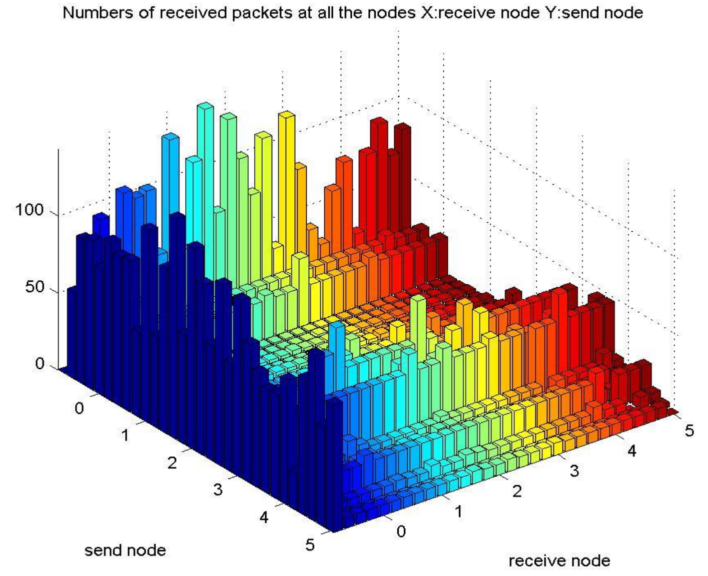

Figure 22: Parameter: Number of received packets

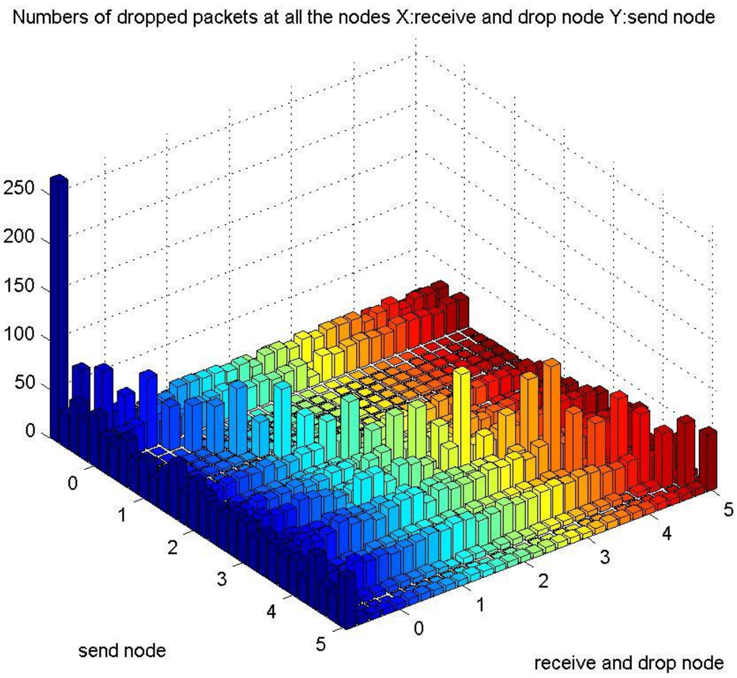

Figure 23: Parameter: Number of dropped packets

From the above three demonstrations we can infer the ratio of dropped packets to that of send and received and ratio of received packets to generated. These illustrations convincingly picture the efficiency of the BCAMAC protocol.

Following is the snapshot of the animation demonstrating the protocol.

Chapter 5

CONCLUSION AND FUTURE SCOPE

WSN is lending ubiquitous computing and communications for an inter-mix of heavy traffic. It has its origin in military and constantly developing history. It is managed at three managerial axes; functional areas and management levels. Smart sensor motes are core of any WSN. Sensor network comprises of sensor field, sensor node deployed wirelessly, sink node, a task-manager module, any number of mobile users connected over a backbone of network such as Internet and satellite. MAC is the lower sub-layer of DLL (Data Link Layer) of OSI Reference model of communication over WSN. An efficient MAC protocol is key for providing an extremely efficient WSN. Several MAC protocols have been classified and proposed in the past that try to alleviate the efficiency issues pertaining to WSN. Some of the major MAC protocols are SMAC (Schedule MAC), TMAC, BMAC, BoxMAC, WiseMAC, RIMAC, XMAC etc. Every MAC has still some issues associated with it, some major and some minor. A simulator model of a real-world system is essentially a generalization. Network Simulator is a discrete event packet level simulator target being networking research and it is being widely used for analyzing performance of these protocols. Of all types of MAC, hybrid of MAC and cross layer MAC protocols have been found to best suited for present types of traffic existing and developing in WSN with time. The scope of WSN lies in the vision "Embed anywhere, Network everywhere economically". Initial goal of deploying a WSN was unsupervised operation with no maintenance. But the requirement today is to serve the vision "Embed anywhere, Network everywhere economically", including the goal. The proposed BCAMAC protocol uses base beacon as the new activation event included in the list to which sender nodes respond. From the previous chapter illustrations, it is clear and evident that BCAMAC decreases end-to-end delay significantly compared to RIMAC and TMAC. WSN is challenging from research point of view because they are severely energy constrained, limited in energy sources, self-organizing and self-healing, scalable and arbitrarily large in number, devices with varied capabilities, remotely deployed. They are required to be adjustable to operating conditions and changes in application requirements, provide security and privacy, handle potentially sensitive information, face hostile environments. There is wide range of smart applications still to be developed and implemented in most parts of the world, where sensors can directly exchange information with entities on Internet: reaching for instance some home environment, creating “network of things”, “tangible Internet”, “Internet of Things” etc. Many enhancements are yet to be implemented in several areas. For instance, commercial applications including “Security of Intra-Car”, “Smart Parking”, “Event Detection” “Vehicular Telematics”, “Structural Health Monitoring” etc. have still a wide scope of development. Research issues in environmental applications include “Greenhouse Monitoring”, “Habitat Surveillance” etc. (Gilbert, 2012). Therefore a efficient MAC protocol has great scope, no matter how many and how better are proposed because speed has no limit.

REFERENCES

[1] A Roy and N Sarma (2010), “Energy saving in MAC layer of wireless sensor networks: a survey”, proc. NWDAA, Tezpur University.

[2] Abayomi M.Ajofoyinbo (2013), “Energy Efficient Packet Duration Value based MAC protocol for wsn”, proc.

[3] Abul Kalam Azad, M.Humayun Kabir, Md. Bellal Hossain (2013), “A survey on schedule-based MAC protocol for wireless sensor network”, proc. IJCSN Vol. 2, Issue 6.

[4] Balvinder Kaur Thind (2004), “A receiver-initiated MAC protocol with enhancements for multi-hop wireless networks”, proc. Thesis report at Washington State University.

[5] Bram Kersten, Chidiebere G.U. Okwudire (2009), “MAC layer energy concerns in WSN: Solved”.

[6] Cesare Alippi, Giuseppe Anastasi, Mario Di Francesco (2009): “Energy Management In wireless sensor network with energy-hungry sensors”, Italy University of Pisa, Italy.

[7] Cristina Cano, David Malone, Boris Bellalta, Jaume Barcelo (2006), “On the improvement of receiver-initiated MAC protocols for WSN by applying schedule”.

[8] D.Antony Pradeesh, B.R. Shyamala Devi (2014), “Duty cycle scheduling based on AMAC protocol for WSN”, proc. International Journal of Science and Research.

[9] David Moss, Philip Levis (2007), “BoX-MACs: Exploiting physical and link layer boundaries in low power networking ”, proc. Technical report.

[10] Edwin Prem Kumar GIlbert, Baskaran Kaliaperumal, Elijah Blessing Rajsing (2012): “Research Issues In Wireless Sensor Network: A survey” , Proc. International Journal of Information And Electronics, Vol. 2, No. 5 .

[11] Gavin Holland, Nitin Vaidya, Paramvir Bahl (2001), “A rate-adaptive MAC protocol for multi-hop wireless network”, proc. Conf. on Mobile Computing and Networking, Italy.

[12] Gholam Hossein Ekbatanifard, Reza Monsefi, Mohammad H.Yaghmaee M., Seyed Amin Hosseini. S. (2012), "Queen-MAC: A quorum-based energy-efficient MAC protocol for wsn", proc. SciVerse ScienceDirect.

[13] H.Wu and Y.Pan (2008), "Medium Access Control in Wireless Networks, Volume II", published by Nova Science Publishers.

[14] Hao Liang (2011), "Double-loop receiver-initiated MAC for cooperative data dissemination via roadside WLAN", proc. IEEE.

[15] I.F.Akyildiz (2001), "Wireless Sensor network: a survey In: Broadband and Wireless Networking Laboratory, GIT, USA.

[16] Jing Ai, Jingfei Kong, Damla Turgut (2011), "An adaptive coordinated medium access control for WSN", proc.

[17] Kyle Jamieson, Hari Balakrishnan, Y.C.Tay (2006), "Sift: A MAC protocol for event driven WSN", proc. Springer.

[18] Linnyer B. Ruiz, Thais R.M. Bragaa, Fabricio a.Silva. Helen p.Assunca (2005) : "On the design of Self-managed wireless sensor network", Proc. IEEE Communication Magazine Federal University of Minas Gerais.

[19] Masanori Takata, Masaki Bandai, Takashi Watanabe (2006), "A receiver-initiated directional MAC protocol for handling", proc. IEEE.

[20] Nuwan Gajaweera, Dileeka Dias (2008), "FAMA/TDMA hybrid MAC for WSN", proc. IEEE.

[21] P.Raja, P.Dananjayan (2014), "Game theory based energy efficient hybrid MAC protocol for lifetime enhancement for wsn", proc. Iranian Journal of Electrical and Electronic Engineering.

[22] P.Udayakumar, Ranjana Vyas, O.P Vyas (2012), "Token Bus Based MAC protocol for WSN", proc. International Journal For Computer Applications.

[23] Protocol design and implementation for wireless sensor network (2008), Piergiuseppe Di Marco's Master degree project, Sweden.

[24] S.Mehta, K.S.Kwak (2010), "HMAC: A hybrid MAC protocol for WSN", proc. IJCNC.

[25] Sana Ullah, Muhammad Imran, Mohammed Alneum (2014), "A hybrid and secure priority-guaranteed MAC protocol for wireless body area network", proc. IJDSN.

[26] Shuguo Zhuo, Zhi Wang, Ye-Qiong Song, Zhibo Wang, Luis Almeida (2010), “iQueue-MAC: A traffic adaptive duty-cycled MAC protocol with dynamic slot allocation”, proc.

[27] Sumita Nagah, Arvind Kakria (2013), “Hybrid MAC protocols for wsn”, proc. IJETCAS.

[28] Surjya Ray, Ilker Demirkol and Wendi Heinzelman (2014), “ADV-MAC: Advertisement based MAC protocol for wireless sensor network”.

[29] T Saranya Lakshmi, S.Jeevitha (2014), “Energy usage reduction in wireless sensor network using asynchronous duty cycle MAC protocol”, proc. IJEETC, vol.3.

[30] Tao Zheng, Sridhar Radhakrishnan, VenkateshSaranga (2005), “PMAC: An adaptive energy-efficient MAC protocol for WSN”, proc.

[31] The evolution of wireless sensor network (2013), Silicon Labratories Inc.

[32] Yanjun Sun, Omer Gurewitz, David B.Johnson (2008), “RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor network”.

[33] Yasmeen Sultana, Abdul Mubeen Mohammad (2014), “Energy efficient routing and fault tolerant clustering using EEPW-MAC protocol”, proc. International Journal of Electrical, Electronics and Data Communications.

[34] Youngmin Kim, Hyojeong Shin, Hojung Cha (2008), “YMAC: An energy-efficient multi-channel MAC protocol for dense wsn”, proc. International Conference on Information Processing in Sensor Network.

[35] Zaher Merhi, Mohamed Elgamel, Samih Abdul-Nabi, Amin Haj-Ali, Magdy Bayoumi (2013), “EVAM-MAC: An event based medium access control for wireless sensor networks with multi-hop support”, proc. IJCSI.

[36] Zhenzhen Liu, Itamar Elhanany (2009), “RLMAC: a reinforcement learning based MAC protocol for WSN”, proc. Inderscience Enterprises Ltd.

Websites Referred:

1. **http://www.isi.edu/nsnam/ns/tutorial**
2. **http://ns2blogger.blogspot.in/**
3. **http://en.wikipedia.org/wiki**
4. **http://people.idsia.ch/~andrea/sim/simnet.html**

APPENDIX

ABBREVIATIONS

1. ALOHA: Abramson's Logic of Hiring Access
2. BEB: Binary Exponential Back-off
3. BMAC: Berkeley MAC
4. BSS: Balanced-energy Sleep Scheduling
5. CAP: Contention Access Points
6. CCP: Coverage Configuration Protocol
7. CMAC: Convergent MAC
8. CMOS: Complementary Metal Oxide Semiconductor
9. CSMA: Carrier Sense Multiple Access
10. DARPA: Defense Research Projects Agency
11. DCF: Distributed Coordination Function
12. DE-MAC: Distributed Energy-aware MAC
13. DMAC: Data Gathering MAC
14. DSN: Distributed Sensor Network
15. DWEHC: Distributed Weight-Based Energy-Efficient Hierarchical Clustering
16. E-LEACH: Enhanced LEACH
17. EMAC: EYES MAC
18. EM-MAC: Efficient Multichannel MAC
19. ER-MAC: Emergency Response MAC
20. EVAM: Event based MAC protocol
21. FAMA: Floor Acquisition Multiple Access
22. GloMoSim: Global Mobile Information System Simulator
23. HCF: Hybrid Coordination Function
24. HEED: Hybrid Energy Efficient Distributed Protocol
25. HIT: Hybrid Indirect Transmission
26. HyMAC: Hybrid MAC
27. IHMAC: Intelligent Hybrid MAC
28. LDAS: Lightweight Deployment Aware Scheduling
29. LDS: Linear Distance-based Scheduling
30. LEACH: Low Energy Adaptive Clustering Hierarchy

31. LMAC: Lightweight Medium Access Control
32. LPL: Low-Power Listening
33. MAC: Medium Access Control
34. MATLAB: Matrix laboratory
35. MC-LMAC: Multi-Channel LMAC
36. MEMs: Micro Electro Mechanical Systems
37. MMSN: Multi-frequency Media Access Protocol
38. MOCA: Multimedia over Coax Alliance
39. MSNL: Maximization of Sensor Network Life
40. NAM: Network Animator
41. NAV: Network Allocation Vector
42. OGDC: Optimal Geographic Density Control
43. OPNET: Optimized Network Engineering Tool
44. OptSim: Optimisation and Simulation
45. OSI: Open System Interconnection
46. PCF: Point Coordination Function
47. PEAS: Probing Environment and Adaptive Sensing
48. PECAS: Probing Environment and Collaborating Adaptive Sleeping
49. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol
50. PEGASIS: Power-Efficient Gathering in Sensor Information Systems
51. PHMAC: Parallel and Hierarchical MAC
52. PW-MAC: Predictive Wake-up MAC
53. QMAC: Quorum-based MAC
54. QoS: Quality of Service
55. RIS: Random Independent Scheduling
56. RTS-CTS: Request-to-send – Clear-to-send
57. RTWAC: Radio-triggered wake-ups with addressing capabilities
58. SAMAC: Sectored-Antenna Medium Access Control
59. SCPMAC: Schedule Channel Polling MAC
60. SERAN: Semi-Random Communication protocol
61. SMAC: Sensor MAC
62. SMACS: Self-organizing MAC for Sensor Networks
63. SNGF: Stateless Geographic Non-deterministic Forwarding

64. SOSUS: Sound Surveillance System
65. SPIN: Sensor Protocols for Information via Negotiation
66. STEM: Sparse Topology and Energy Management
67. TCL: Tool Command Language
68. TMCP: Tree-based Multichannel MAC
69. TRAMA: Traffic-Adaptive Medium Access
70. ZMAC: Zebra MAC