An Energy Efficient Cross Layer Design Scheme for Wireless Sensor Networks

by

Srajan Raghuwanshi

Submitted to the Department of Electrical and Computer Engineering at

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

in partial fulfillment of the requirements for the degree of

Master of Science

Committee Members:
Dr. Amitabh Mishra, Thesis Advisor, Advisory Committee
Dr. Sandeep K. Shukla, Advisory Committee Member
Dr. Ira Jacobs, Advisory Committee Member

August 29th 2003

Blacksburg, Virginia

Keywords: MAC, Wireless Sensor Networks, Energy, Cross layer

An Energy Efficient Cross Layer Design Scheme for Wireless Sensor Networks

Srajan Raghuwanshi

(ABSTRACT)

Wireless Sensor Networks (WSNs) are wireless networks that have recently drawn significant research attention since they offer unique benefits and versatility with respect to sensing, allowing low-power and low-cost rapid deployment for many applications that do not need human supervision. WSNs are self-created and self-organized by the collection of a large number of sensor nodes interconnected by multi-hop wireless paths. The sensor nodes are network embedded systems with Integrated Chips (ICs) to allow signal processing and micro-sensing. Each wireless sensor node is a micro-electro-mechanical device and can only be equipped with a limited power reserve. While energy consumption occurs in sensing, data processing and communications, care should be exercised to make the most of the expendable power source for the node.

Though considerable research is being done in the area of energy saving techniques for WSNs, most of the proposed techniques have focused on energy awareness at different network layers in WSNs. Furthermore, most of the proposed techniques are based on protocols for mobile ad hoc networks that do not look into the possibility of a cross-layer design strategy that can exploit the unique features of WSNs. There still exists the need for a universal protocol that can be applied to such networks in general. In this thesis, we focus such a research on optimizing the energy consumption by suggesting a novel cross-layer architecture at the network/data-link layer for sensor networks. We have developed a scheme for better and improved energy efficiency in WSNs by combining the ideas of energy-efficient cluster formation and medium access together. Our cross-layer scheme provides good performance in terms of WSN-lifetime, scalability and minimizing network-wide energy consumption. The scheme is based on a collaborative approach supported by formation of dynamic clusters functioning with a traffic aware MAC (medium access control) scheme. Our MAC scheme incorporates a self-learning, traffic adaptive algorithm for varying traffic conditions inherent to the WSNs. The design methodology and results in this thesis aim at producing a reliable and scalable energy-aware sensing network, in spite of node failures, minimizing energy consumption at the same time.

Acknowledgements

I would like to thank my professor, Dr. Amitabh Mishra, for being a wonderful advisor. Dr. Mishra has been a great help and has given me lots of support with timely encouragement for a continuous and constructive thought process. I thank him for his categorical comments on the thesis; it has helped me improve my writing skills and present my research in a much better way. I would like to thank Dr. Shukla and Dr. Jacobs for guiding me towards the end of my thesis and defense period. I am grateful to my committee chair for their time in reviewing this document recursively over much iteration, and suggesting technical modifications and corrections.

I am indebted to all my friends and family who made it possible for me – to achieve what I have today.

Table of Contents

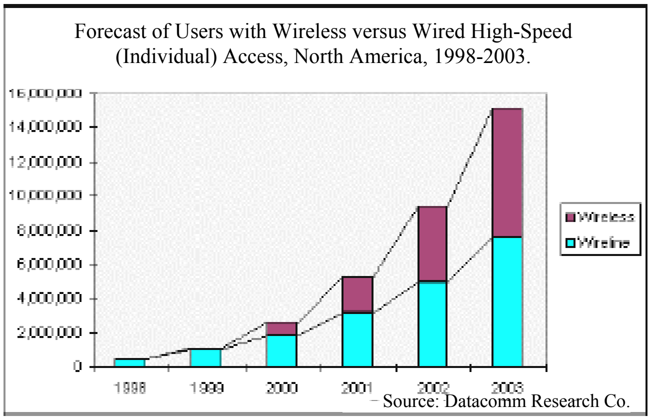
List of Figures

List of Tables

1. Introduction

Over the past few years we have seen a tremendous growth in wireless-based networks and systems that has made the wide-spread commercial use of wireless devices - cell phone, global positioning systems (GPS), pagers - possible. The rapid advancement made in the industry since the advent of wireless-based technology has led some research-based studies to believe and predict [30] that wireless access in laptops, personal digital assistants (PDAs), mobile phones would overshoot wired access by later next year [Figure 1.1]. Furthermore recent advances in processor, memory and radio technology [43] coupled with the advancement in the wireless technology has allowed portable devices to support several important wireless applications, including real-time multimedia communication [32], medical applications, surveillance [33, 34] using sensor networks.

Sensor network is a developing domain with operational demands unlike any other recent paradigm in wireless communication. Every functional specification of wireless sensor networks requires it to be different from the current wireless technology – sensor node densities being extraordinarily high, very low data rates, extremely low transmission and reception powers at the nodes, micro-sized nodes with expected sensor node lifetimes between 5-10 years on battery power. However, amidst the afore-mentioned specifics for the revolutionary wireless domain of wireless sensor networks lies the challenge of designing a system that can optimally incorporate all the functional specifications. While cheaper silicon and circuit complexity has allowed the realization of Moore's law allowing Integrated Circuits (ICs) complexity to double consistently over 18 months at an almost constant cost, Moore's law just seems inapplicable to apply to batteries. Depending on the particular chemistry of batteries and prolonged refinement of their constituents, energy densities of batteries has only doubled roughly over a time span of 5-20 years [10]. The tremendous potential for rapid deployment of such networks is thus offset by their limited power reserves attributed to their minute size.


Forecast of Users with Wireless versus Wired High-Speed (Individual) Access, North America, 1998-2003.

Source: Datacomm Research Co.

Figure 1.1: Predicting the usage of wireless access in comparison to wired access [reported by Datacomm Research Co.] [31]

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) represent a new paradigm in wireless technology drawing significant research attention from diverse fields of engineering. WSN technology is stated to be one of the "***10 Emerging technologies** that will change the world*" [35]. The tremendous potential for rapid deployment of such networks is underway with busy researchers creating and optimizing WSN technology all around the world. The vision of many researchers is to create sensor-rich "smart environments" [36] through planned or ad-hoc deployment of thousands of sensors, each with a short-range wireless communications channel, and capable of detecting ambient conditions such as temperature, movement, sound, light, or the presence of certain objects.

WSNs can be deployed extensively in the physical world and spread throughout our environment. They can be sited far from the actual occurrence and can still be used for data aggregation and collection from a remote location far away from the phenomenon. The WSNs comprise of a large number of application-specific wireless sensor nodes (typically in hundreds of thousands in number) spread over varying topographies. This kind of random placement of the sensor nodes does not follow any fixed pattern and the density of nodes is not dependent on any factor. Once they are deployed in the environment (under scrutiny where sensing needs to take place), these hundreds and thousands of nodes have to organize themselves in the network by listening to one another. They self-organize themselves by creating multi-hop wireless paths through mutual co-operation. The nodes work collectively and collaborate together on common tasks of sensing/data-collection/communications etc to provide good network-wide performance in terms of network life-time, latency, and uniform density of available nodes for sensing.

WSNs offer unique benefits and versatility with respect to low-power and low-cost rapid deployment for many applications that do not need human supervision. Some of these applications include disaster recovery, military surveillance, health administration, environmental & habitat monitoring, target-tracking etc. Due to the large numbers of nodes involved in the WSN deployment new benefits to the afore-mentioned sensing applications including [28]:

- Extended range of sensing - WSNs enable large numbers of nodes to be physically separated; while nodes located close to each other will have correlated data (e.g., these nodes will be collecting data about the same event), nodes that are farther apart will be able to extract information about different events.
- Robustness and fault-tolerance - Ensuring that several nodes are located close to each other and hence having correlated data makes these systems much more robust in terms of data sensing (even though it involves redundancy). In case of WSNs even if a small number of sensor nodes from a network fail, there is enough redundancy in the data from different nodes that the system may still produce acceptable quality information.
- Improved accuracy - While an individual sensor's data might be less accurate than another independent sensor's data (both sensor nodes are assumed to be in close

proximity to the detected event) in the WSN, combining the data from nodes increases the accuracy of the sensed data. Since nodes located close to each other are gathering information about the same event, aggregating their data enhances the common signal and reduces the uncorrelated noise as well.

- Lower cost - Due to reduced size, reliability, and accuracy constraints on sensor nodes, these nodes are much cheaper than their high-accuracy high-complexity sensor counterparts.

However to be able to realize all the discussed specifications we need to design protocols that can provide appropriate support and allow the wide-spread use of WSNs.

1.2 Wireless Sensor Node

1.2.1 Node Architecture

A sensor is any device that has the capability of measuring a physical quantity from the place of its deployment and converting it to a quantitative measure that can be interpreted by the system that the sensor is embedded into. When such a sensor is embedded onto a system that can network itself wirelessly with other systems using standard network interfaces based on a *similar* protocol stack as shown in Figure 1.2 we have a wireless sensor node. We compare the protocol stack in sensor network architecture to be similar to the seven-layer OSI (Open Systems Interconnect) of conventional network architecture as there is no universally accepted protocol stack in WSNs. However, in our discussion of protocol-related issues in Chapter 2, we elaborate on the specific protocol stack for WSNs.

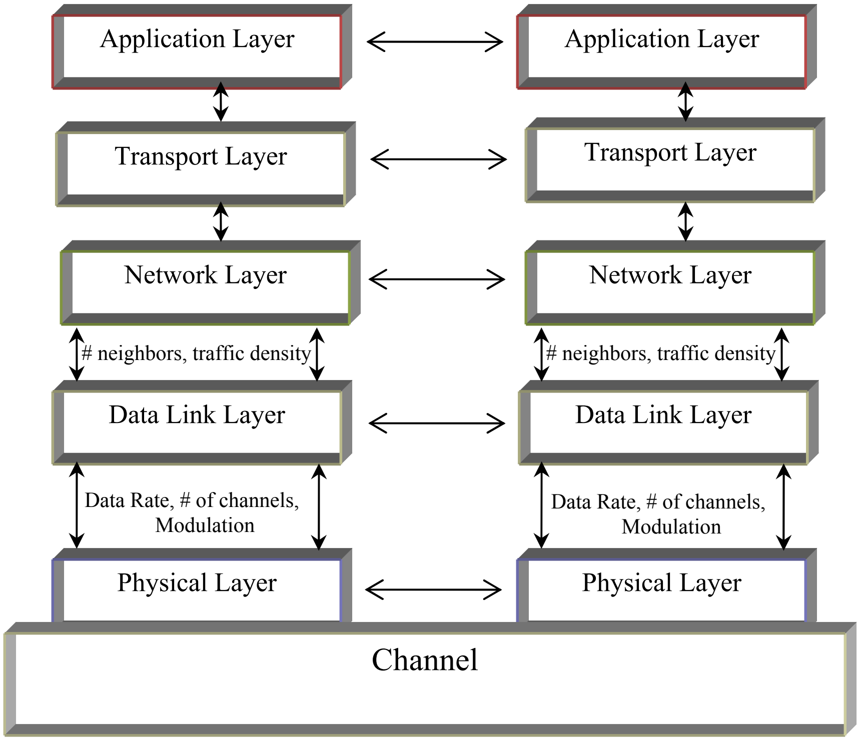


Figure 1.2: General Protocol structure of sensor nodes in WSN

Sensor nodes can be deployed over a variety of geographic topographies for example, shown as a part of Figure 1.3 with 100 nodes (generated using the SensorSim simulator [44]). The architecture of the individual sensor node, however, is solely dependent on the nature of their application(s)-specific deployment. The generalized architecture [15] of such an individual node – a Micro-electro-mechanical system (MEMS) based node – is illustrated in Figure 1.3. Each sensor node consists of a sensing unit, a processing unit, a communication unit, and a power source. The sensor and the ADC together form the sensing unit, where the analog signals produced by the sensors are converted to digital signals by the ADC, before being supplied to the processing unit (comprising the processor and storage). The processing unit is responsible for carrying out the allotted sensing tasks. The communication unit acts as the link between the sensor node and the network. The power scavenging source generally derives energy from solar, battery power. A more qualitative description of each of these units is as follows,

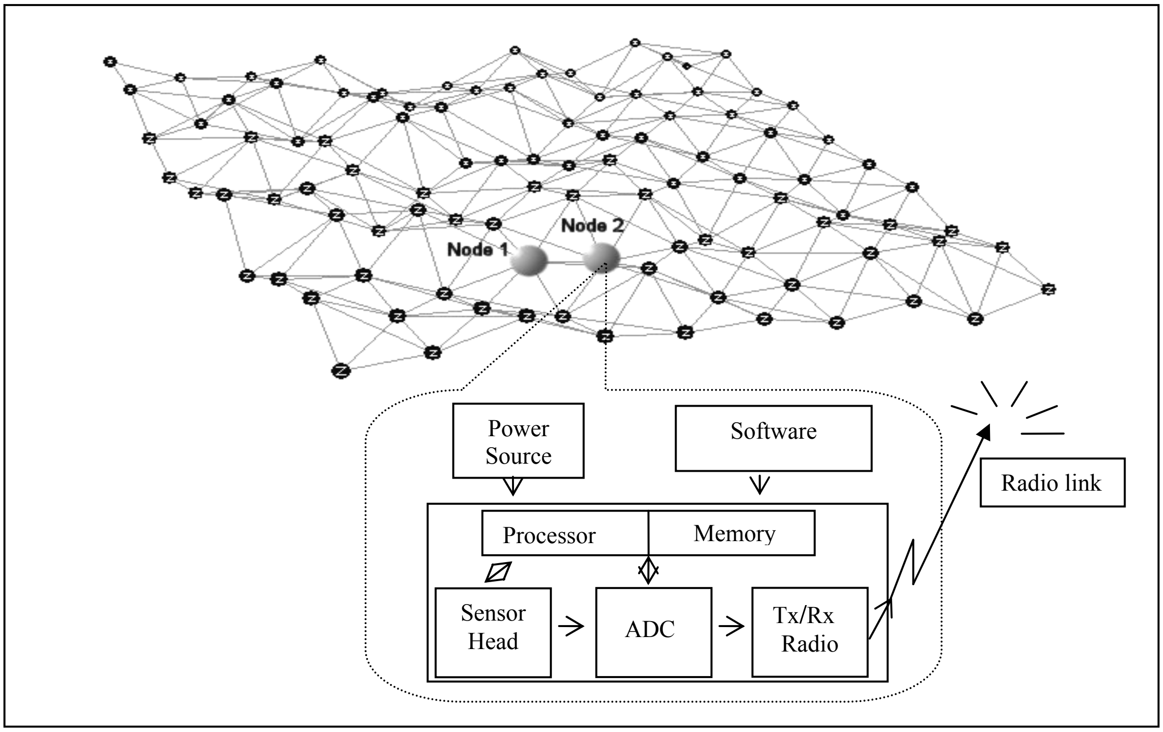


Figure 1.3: Individual node architecture in a WSN containing hundreds of thousands of nodes that collaborate for data collection and processing.

- Processing unit – consists of a microprocessor responsible for the control of the sensors and execution of communication protocols. The μ-processor usually operates under various operating modes for power management purposes. But shuttling between the various operating modes involves consumption of power, so the energy consumption levels of the various modes should be considered while looking at the battery lifetime of each node.

- Communication unit – A significant amount of energy is used by the radio during communication in the WSN. This necessitates the use of typical low power RF (radio-frequency) radios that are capable of delivering a fair bandwidth in a single channel on the ISM band. Radios can operate under the Transmit, Receive, Idle and Sleep modes. It is important to completely shut down the radio rather than put it in the Idle mode when it is not transmitting or receiving because of the high power consumed in this mode.
- Sensing unit – consists of a group of sensors and actuators and link the node to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance which is not required.
- Power source – consists of a battery that supplies power to the node. Usually the rated current capacity of a battery being used for a sensor node is lesser than the minimum energy consumption required leading to the lower battery lifetimes. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often.

Examples and the current most popular choices of sensor nodes in building WSNs are listed and described in Appendix I.

1.2.2 Energy related issues

Energy consumption is the most important factor in determining the life of a WSN as the sensor nodes comprising the network are all battery-powered and thus limited by very low energy resources for tasks involving data sensing, processing and communications. Energy optimization techniques therefore take precedence in WSNs to allow preservation of energy for prolonged network life-times. This can be done by considering energy awareness issues in every aspect of design and operation of each sensor node. Energy saving protocols and techniques also need to be addressed for collective groups of communicating sensor nodes in order to have better overall performance and improved energy efficiency in the entire WSN.

The lifetime of a sensor network can be increased significantly if the operating system, the application layer and the network protocols are designed to be energy aware. The power consumed by the sensor nodes can be reduced by developing design methodologies and strategies that support lower energy wastage. Power management in radios is also a very important issue because radio communication consumes a lot of energy during operation in comparison to the overall energy consumption of each node in the WSN as a whole. Methods that can optimally require the radios to sleep for longer periods of time when no event is detected should also be an area of focus. Related work is discussed in Chapter 2.

Another aspect of sensor nodes that involves increased battery consumption is when they behave as routers, when data is routed to central points that are connected to, perhaps a larger processing infrastructure. Intelligent radio hardware can be used to identify and redirect packets that need to be disseminated to such data aggregation points. This would also reduce the computing overhead and processing times in the WSN so that the intermediate nodes would need to only forward the unprocessed data packets to the central points for combined data processing and dissemination. Traffic can also be distributed to maximize the life of the network by researching methods on distributing traffic more uniformly throughout the network. Methods that discourage using the same path for routing traffic/packets continuously regardless of how much energy is saved should also be looked into for better fault-tolerance ands robustness. Such methods are required because frequent rupturing in the connectivity of the network due to failure/sleep-schedules of participating nodes can cause sensor nodes that are in close proximity of the detected event to not allow dissemination of data within various sections of the WSN.

Thus, care should be observed to make the most of the expendable power source for the node.

1.3 Meeting the Design Goals

In scenarios where the sensors are operating in remote or dangerous (hostile/unfavorable) territory, it may be impossible to retrieve the nodes in order to recharge batteries. In other sensor network scenarios, such as machine monitoring or medical monitoring, it may just be inconvenient to replace the battery of a node. Therefore, the network should be considered to have a certain lifetime during which nodes have energy and can gather, process, and transmit information. This means that all aspects of the node, from the sensor module to the hardware and protocols, must be designed to be extremely energy-efficient. Decreasing energy usage by a factor of two can double system lifetime, resulting in a large increase in the overall usefulness of the system. In addition to reducing energy dissipation, protocols should be robust to node failures in order to maximize system lifetime. The protocols should be fault-tolerant, such that the loss of a small number of nodes does not greatly affect the overall system performance. In addition, the protocols should be scalable such that the addition of new nodes requires low overhead to incorporate the nodes into the existing network.

Events occurring in the environment being sensed may be time-sensitive. Therefore, it is often important to bound the end-to-end latency of data dissemination. Protocols should therefore minimize overhead and extraneous data transfers. Furthermore, for sensor networks, the end-user does not require all the data in the network because

> (1) data from neighboring nodes are highly correlated, making the data redundant, and
> (2) end-user cares about a higher-level description of events occurring in the environment the nodes are monitoring.

The quality of the network is therefore based on the quality of the aggregate data set, rather than the quality of the individual data signals; protocols should be designed to optimize for the unique, application-specific quality of a sensor network.

To summarize, wireless sensor network protocols should be:

- self-configuring, to enable ease of deployment of the networks,
- scalable algorithms to support very high scalability of sensor nodes,
- energy-efficient and robust, to extend system lifetime, and
- latency-aware, to get the information to the end-user as quickly as possible.

The work presented in this thesis focuses on methods that try to incorporate all of these features.

1.4 Contributions of this research

We have designed and implemented protocol architecture for WSNs that achieves low energy dissipation and higher scalability. Since sensor-heads can be replaced as single modules on every single sensor node (in Figure 1.3) for serving the purpose of specific applications, our research looks at a cross-layer (at the network/data-link layer) design for a very general application scenario. Specifically so, by working on the protocol architectures for individual nodes we have been able to design a robust, energy-efficient, and scalable scheme for WSNs that can be applied to any generic application that WSNs may be used for. The results of our research show that greater amount of energies can be saved by using the cross-layer design. Our cross-layer scheme provides good performance in terms of

- Scalability,
- Network wide life-time, and
- Minimizing network-wide energy consumption

The scheme is based on a cross-layer collaborative approach supported by formation of dynamic clusters functioning with a traffic aware MAC (medium access control) scheme. Our MAC scheme incorporates a self-learning, traffic adaptive algorithm for varying traffic conditions inherent to the WSNs. The titles and the list of technical manuscripts that have been accepted at various technical conferences as a part of the study are listed below,

1. "**An Approach Towards Improved Energy-efficiency in Wireless Sensor Nets**", IEEE Radio and Wireless Conference 2003, RAWCON 2003, August 10-13, Boston MA.
2. **"A self-adaptive clustering based algorithm for increased Energy-efficiency and Scalability in Wireless Sensor Networks**", Proceedings IEEE Vehicular Technology Conference 2003, VTC 2003, October 6-9, Florida.
3. **"Scalable Energy Efficient Sensors**", Fact sheet for Oak Ridge National Laboratory (ORNL).

1.5 Thesis Organization

This thesis begins with a detailed discussion of general-purpose protocol architectures for wireless networks, including link-layer, media access control (MAC), protocols (Chapter 2). We discuss related background and present a literature survey of available energy-saving schemes at the data-link layer in Chapter 2. Chapter 3 discusses our approach to the problem by elaborating on the proposed research. In this chapter we state our assumptions followed by a detailed outlook to our proposed methodology. Chapter 4 follows with simulation models. We discuss our results and their analyses in Chapter 5. The thesis ends with conclusions and a discussion of future directions in Chapter 6. We have also included an Appendix that includes information on current sensor hardware with photographs of real sensor nodes. The appendix also includes some discussion on various traffic distributions discussed in context of our work.

2. Background

Energy-awareness is a well-studied problem, with a rich history stretching back to the advent of the first few computer wireless networks [37]. However due to the unique design requirements in WSNs virtually every aspect of the system's design can be considered for energy saving techniques [6]. In this chapter we however, limit the scope of our background literature search on select work based on energy saving mechanisms at the data link-layer. We go over the available methods and point out their unique functionalities explaining various proposed schemes in the literature. We also try to qualitatively compare the various available methods towards the end of the chapter.

2.1 Protocol Design Issues

WSNs are generally deployed in resource-constrained environments where the longevity and sustenance of the WSN is dependent on the co-operation of all the nodes in the neighborhood as well as the different layers in the protocol stack (shown in Figure 2.1) of each of the sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, and physical layer. In conjunction with the various layers in WSNs the sensor nodes use power management to use available power. Scheduling the sensing tasks to specific nodes within the site of occurrence is managed by Task management. Factors that influence task management include the status of varying power levels of nodes in the WSN so that not all sensor nodes in a given region are required to sense. Mobility management is required to detect and register the movement of sensor nodes so that information of neighboring nodes is kept up-to-date. Networking within the umpteen sensor nodes is possible due to these management features incorporated within the WSN protocol stack that allow sensor nodes to work collaboratively, route data efficiently, and share the limited resources most optimally.

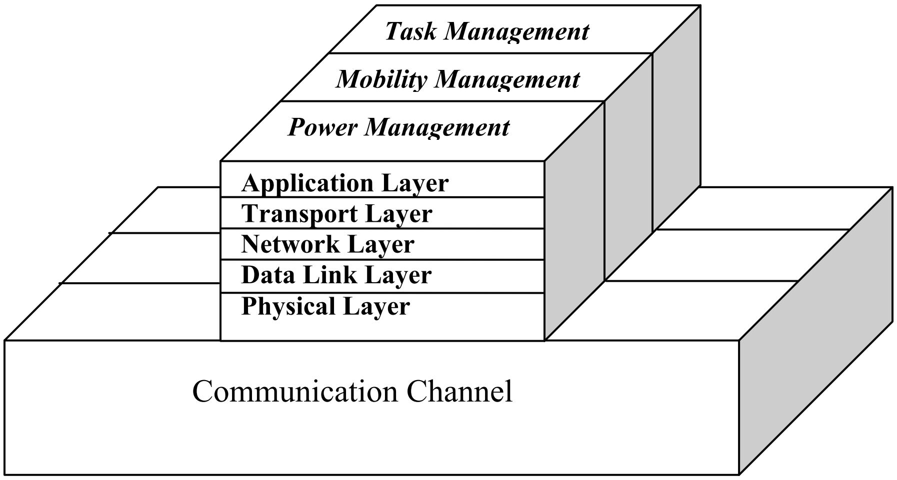


Figure 2.1: WSN protocol stack incorporated with task-, mobility- and power-management.

Based on the sensing application that the WSN is being deployed for, application software can be built and used for the application layer. Based on the requirement of the sensing application/task the transport layer provides a reliable flow of data from transmitting node to receiving node. The network layer is concerned with the routing of data from the transport layer and the data link layer addresses methods on achieving a reliable, efficient communication between two communicating nodes. Factors such as propagation delay, energy awareness, medium access and error control are managed within the data link layer. Finally the physical layer [2] addresses issues responsible for frequency selection, signal detection, modulation and encryption.

While most of the above-mentioned entities would need to be handled rather seriously in existing networks, WSNs do not need to address all the attributes for their successful use. As traditional networks may aim to achieve high quality of service (QoS) provisions, sensor network protocols must focus primarily on power conservation due to limited, generally irreplaceable, power sources that can decide the longevity of the deployed WSN. Among the protocol layers discussed above the link-layer entities must specifically sense the environment and signal it to the upper layer entities for appropriate adaptation to adjust for link availability, link capacity, latency, congestion status and existing error conditions.

A few issues discussed here explain the importance of the need to design a robust data-link layer [24]. When there is no event, the entire system including the data link layer should hibernate. The data link layer should use distributed methods so that clear and frequent communication across sites, achieving coordinated and standardized measures becomes possible. Furthermore, a distributed network is more scalable and more robust because it has no infrastructure. Global synchronization must not be required in the data link layer. Since large numbers of sensor nodes are deployed densely in a WSN, neighboring nodes stay very close to one another making short distance one hop communication a clear choice so that lesser power can be consumed. The limited bandwidth of wireless channels between nodes, low-power consumption requirements further aggravates the situation, as message overheads need to be kept at a minimum level over the single-hop communications. They must have inbuilt tradeoff mechanisms that give the end user the option of prolonging network lifetime at the cost of lower throughput or higher transmission delay. Power is consumed only if a "data-centric" [3] event occurs. Finally, the data link layer design needs to be simple and robust.

2.2 Motivation

The Medium Access Control (MAC) supported in the data link layer is an essential functionality that can be exploited for increased energy saving. The MAC in general is responsible for placing data on the transmission media, deciding when to listen and when to talk, and handling fixed-length parts of packets in the network. The MAC is also responsible for sharing the limited communication resources fairly and efficiently between the nodes. MAC in sensor networks however needs to be very different from the traditional networks because of its constraints on computational ability, storage and energy resources. Therefore medium access control should be energy efficient and should help with a fair utilization of the bandwidth among all the nodes in the network. In sensor networks, the primary objective is to sample the residing environment for information and send it to a higher processing infrastructure (base station) after processing it. The data traffic may be low for lengthy periods with intense traffic in between for short periods of time [12]. Most of the time, the traffic is multihop and heading towards some larger processing infrastructure. At each of the nodes, there is traffic originating out of the node and traffic being routed through the node - because most nodes act as both data sources and routers. There are several restrictions on sensor nodes too. They can have little or no dedicated carrier sensing (CS) or collision detection (CD) embedded as a part of their hardware (unlike the case in traditional network hardware where additional circuitry is used for CS/CD [30]). Based on application specific they usually have no specific protocol stacks, which could specify the design of their media access protocol.

2.3 Sensor MAC Protocol Requirements

There have been many proposed MAC protocols in the literature specific to ad-hoc wireless networks. The proposed MAC channel access methods in ad hoc networks can be broadly divided into two categories: contention based and organized methods as outlined in [1]. Even though modifications have been suggested to the proposed ad hoc protocols with reference to WSNs, they are all not well suited for WSNs because of unique functionalities. To illustrate the reason a few differences are outlined here

- WSNs have very limited power, and processing capability in comparison to ad-hoc networks.
- The numbers of nodes in the WSN outnumber the nodes in ad-hoc networks by very large numbers.
- In WSNs the frequent changes in the topology due to the nature of operation of nodes gives rise to a dynamic topology [22].
- WSNs mostly use a broadcast communication model as compared to the point-to point communication paradigm used in ad-hoc networks [45].

The requirements of the sensor network MAC protocol design that differ from the MAC protocol of conventional wireless networks are as follows:

Power efficiency: Radio is one of the most energy consuming parts in a sensor node. The primary sources of energy waste in the radio of a sensor node are collisions, idle listening, overhearing and control packets. Collision causes a packet to be corrupted by another packet. Since this packet is discarded, the energy consumption per successful transmission will increase. Idle listening occurs when the node consumes power listening to the channel for possible traffic even when there is no packet to be received on the channel. Overhearing occurs when a node consumes energy to receive a packet that is not destined to it. Finally, the number of control packets should be minimized to eliminate the energy consumption related to them.

Real-time deadline: Many applications require the guaranteed arrival of sensor data to Access Point (AP) within a specific deadline. One example is *security monitoring*.

Another example may be *traffic light controller*. In traffic light controller, sensor nodes are placed along the road near the traffic light and inform the AP of the number of cars waiting behind the lights along each direction. AP then adjusts the duration of the lights accordingly. Since the traffic light duration is around 30-90 seconds, MAC protocol should be able to guarantee an upper bound on the maximum delay from the sampling of the sensors until the time when data reaches AP.

Congestion control: Congestion control schemes adjust the traffic generation rate based on the feedback from the network. This is necessary in sensor networks so as to eliminate the bottleneck at the sensor nodes near AP since the traffic generated at all nodes pass through a few nodes around AP. The control scheme affects both the delay that each packet experiences and the power consumption per successful transmission. If the packet generation rate is above the rate that the network can handle, the queue length at the nodes on the route to AP increases, which causes an increase in the delay of the packets. Moreover, if the queue length of a node increases too much so that some packets are dropped, energy invested in these packets at each hop from their source to this node is wasted.

Fairness: The aim of most of the sensor network applications is to collect the same amount of data from each node in the field. Receiving higher number of packets from one sensor node at the cost of lower number of packets received from the other one should be eliminated. Therefore, the scheme should maximize the minimum number of packets received from the nodes in the network.

2.4 Sensor MAC Protocol – Proposed Schemes

Now we discuss some of the well known MAC protocols in the realm of WSNs along with a discussion on a few specific protocols for ad-hoc networks that have inspired the existing design of protocols for WSNs. Current MAC protocols for sensor networks can be divided into contention-based and TDMA protocols. We start with the review of previously proposed contention-based protocols and then continue with TDMA-based protocols.

The first class of protocols is contention based protocols. The MAC protocols in this class that provide power efficiency are based on exploiting the absence of traffic in listening state by putting the radio in sleep mode. These protocols differ from each other according to the radio wake-up algorithm. In [21], a separate wake-up radio is used to power down the normal data radio as long as there is no packet transmission or reception, based on the assumption that the listen mode of the wake-up radio is ultra low power. If a neighbor node wants to transmit a packet, it first sends a wake-up beacon over a wake-up channel to trigger the power up of the normal radio and then sends the data packet over the data radio. This protocol is successful in avoiding overhearing and idle listening problems in the data radio, but it is unable to solve the collision problem. Moreover, the difference in the transmission range between data and wake-up radio may pose significant problems.

2.4.1 CSMA Based Protocols

1. **IEEE 802.11** Standard [47] is a multiple access technique based on CSMA/CA (Collision Sense Multiple Access/Collision Avoidance) [48]. While many improvements over the original 802.11 standard (Table 1 below) in terms of bandwidth, speed, modulation-schemes have been incorporated as part of wireless technologies as shown in Table 1 and standards [11], the basic protocol still remains the same. It lets the sender initially sense the carrier or medium to determine if it is idle for use by the sender. If the carrier is busy, the mobile defers transmission and enters the back-off state. The time period following this transmission is called the contention window and consists of a pre-determined number of transmission slots. The mobile, which entered back-off, randomly selects a slot in the contention window. It also continuously senses the medium during the time up to its selected contention slot. If it detects transmission from some other mobile during that period, it enters the back-off state again. If no transmission is detected, it transmits the access packet and captures the medium. The MACA protocol [46], addresses the hidden terminal and exposed node problem in 802.11. Extensions to the basic protocol include provisions for MAC-level acknowledgments and Request-To-Send (RTS)/Clear-To-Send (CTS) mechanisms. The IEEE 802.11 [47] standard recommends the following technique for power conservation. A mobile that wishes to conserve energy may switch to sleep mode and inform the base station of this decision. From then on, the base station buffers packets received from the rest of the network that are destined for the sleeping mobile. The base station periodically transmits a beacon that contains information about such buffered packets. When the mobile wakes up, it listens for this beacon and responds to the base station which then forwards the packets. This approach conserves energy but results in additional delays at the mobile that may affect the quality of service (QoS). It is essential to quantify this delay in the presence of QoS delay bounds for individual connections. A comparison of energy-saving mechanisms in the IEEE 802.11 and HIPERLAN standards is presented in [49].

Table 1: Various modifications suggested to the IEEE 802.11 standard

Standard	Ratified	Band	Speed	Modulation	Comments
802.11	1997	900 MHz	2 Mbps	FHSS	Obsolete
802.11a	1999	5 GHz	Up to 54 Mbps	OFDM	Higher data rates but less transmission distance
802.11b	1999	2.4 GHz	1-11 Mbps	DSSS	Most popular Increased speed from 802.11 of 2 Mbps
802.11g	2003	2.4 GHz	Up to 54 Mbps	OFDM	Higher data rates in 2.4 GHz band Backward compatible with b
802.11e	Focused on improving multi-media transmission & quality of service. Backward compatible.				
802.11f	Focused on enhancing roaming between Access Points and interoperability between vendors.				

2. **PAMAS** (Power Aware Multi-Access Protocol with Signaling [3]) is a multi-access protocol developed for ad hoc radio networks based on the classical MACA protocol [46] proposed for radio networks (please see reference for a detailed study of MACA). Even though PAMAS addresses MAC issues for ad hoc networks, it discusses a unique feature in its medium access that conserves battery power at nodes by intelligently powering off nodes that are not actively transmitting or receiving packets. In this thesis, our traffic-adaptive scheme is inspired by a similar method. The state diagram that forms the basis of PAMAS is shown in Figure 2.2. Six states namely – Idle, AwaitCTS (Await-Clear To Send), BEB (Binary Exponential Backoff), Await Packet, and Transmit Packet – are shown in the state diagram and a node has an equal probability of having any of the six states. When the node is not transmitting or receiving a packet, or does not have any packets to transmit, or does have packets to transmit but cannot transmit due to a neighbor receiving a transmission it is supposed to be in the Idle state. When the node receives a packet to transmit (in the case of WSNs sensor nodes can also generate data based on what the sensor node senses around its vicinity) it transmits a RTS (Request To Send) and enters the AwaitCTS state. The BEB state is used if an awaited CTS does not arrive. If CTS does arrive, then the node enters the Transmit state and begins transmitting the packet to the intended receiver which eventually switches to the Await Packet state after transmitting the CTS. Meanwhile, if the intended receiver does not receive the packet in one round trip time (plus processing time), it returns back to the Idle state. In the case of a packet arrival the node transmits a busy tone over the signaling channel and switches to the Receive Packet state as seen in the state diagram.

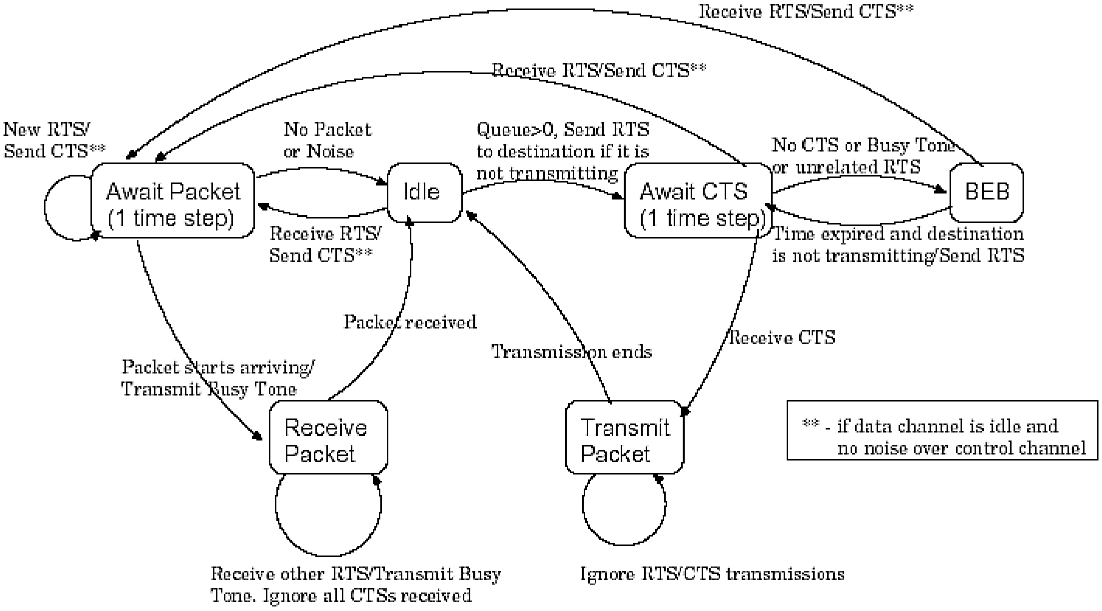


Figure 2.2: State diagram explaining the operation of PAMAS [3]

Though PAMAS avoids overhearing among neighboring nodes by using out-of-channel (inspiring the design of S-MAC [27]), it does not quite address the issue of reducing idle listening. Furthermore, a limitation in its design involves the use of two independent radio channels, which in most cases indicates two independent radio systems on each nodes causing higher complexity and cost.

3. S-MAC (Sensor MAC [27]) – While IEEE 802.11 and PAMAS have been designed specifically for wireless, ad-hoc networks respectively, S-MAC has been designed primarily for WSNs. S-MAC identifies the following main sources of energy wastage in sensor operation :
    - Collision
    - Overhearing
    - Control Packet overhead
    - Idle listening

   S-MAC prevents overhearing by in-channel signaling, using the RTS (Request To Send) and CTS (Clear To Send) packets (similar to the IEEE 802.11 mechanism). This protocol is based on the premise that all the nodes are co-operating for a common task and hence application specific code can be distributed throughout the network. When an interfering node hears a RTS and/or CTS packet, it goes into sleep mode. The timing relationship between the receiver node and different senders is shown in Figure 2.3. In order for a node to receive both SYNC and a data packet, S-MAC divides the listen packets into two parts. The first part is for receiving SYNC packets and the second one is for receiving RTS packets as seen in the figure.

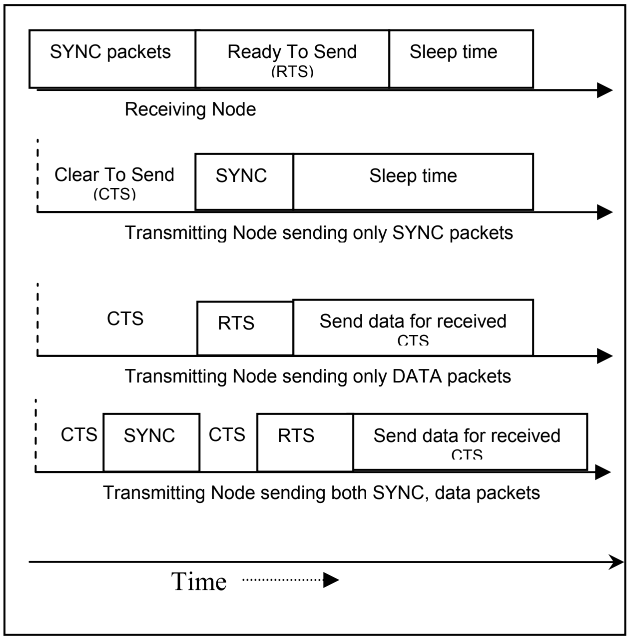


Figure 2.3: Timing relationship in S-MAC

   This protocol also avoids idle listening through periodic listen and sleep modes, the schedules of which are known by neighboring nodes. The problem with this protocol is that it uses RTS/CTS packets to avoid contention. The effect of these control packets on energy consumption is significant when the data packet length is on the order of RTS/CTS packet length (for instance the data packet length is almost equal to RTS/CTS packet length since data packet only contains the place of the spot and whether it is free or not). Moreover, the periodic listen/sleep requires synchronization among neighboring nodes, which uses more control packets. In addition, the latency increases since a sender must wait until the receiver wakes up before it can transmit the packet. Furthermore, per-node fairness is traded off against energy savings.

4. **STEM** (Sparse Topology and Energy Management) [9] protocol trades energy savings for latency through listen/sleep modes as in S-MAC but by using a separate radio. When a node wants to send a packet, it polls the target node by sending wake-up messages over a paging channel. Upon receiving a wake-up message, the target node turns on its primary radio for regular data transmissions. The purpose of using a separate paging channel is to prevent polling messages from colliding with ongoing data transmissions. This scheme is effective only for scenarios where the network spends most of its time waiting for events to happen. Otherwise, the polling through a stream of wake-up messages, collisions and overhearing may cancel out the energy savings obtained by sleep modes.

5. An adaptive transmission rate control mechanism is proposed in [30] to achieve medium access fairness and congestion control. The scheme looks at the following challenges in multihop sensor networks,

    - The originating traffic from a node has to compete with the traffic being routed through that node.
    - An undetected node might exist in the network which might result in unexpected contention for bandwidth with route-thru traffic.
    - The probability of corruption and contention at every hop is higher for the nodes which reside farther away from the higher processing infrastructure.
    - Energy is invested in every packet when it is routed through every node. Therefore, the longer a packet has been routed, the more expensive it is to drop that packet.

    The data origination rate at each node is controlled in order to allow route-thru traffic to propagate. A progressive signaling scheme for route-thru traffic is then used to propagate the back pressure deep down into the network for those nodes to lower their data origination rate. This provides fair allocation of originating and route-thru traffic at each node by eliminating any concession to nodes closer to an Access Point (AP) over those far from AP. In addition, it brings congestion control by avoiding data origination rate to increase above the level that network can handle. Moreover, this scheme achieves good energy efficiency when the traffic is low by maintaining the fairness among the nodes. However, the protocol does not give any real-time delivery guarantee and does not eliminate idle listening and overhearing.

2.4.2 TDMA Based Protocols

The second class of MAC protocols is TDMA-based protocols. The advantages of a TDMA based scheme are elimination of overhearing, collision and idle listening.

However, the currently proposed TDMA protocols are based on performing TDMA scheduling in real communication clusters. The overhead of forming these clusters, and inter-cluster communication and interference may reduce the efficiency of TDMA. Cluster problem can be solved by performing TDMA scheduling for all the nodes in the network by the usage of a simple high power Access Point (AP).

1. **SMACS** (Self-Organizing Medium Access Control for Sensor Networks) [15] protocol aims to achieve power conservation based on TDMA-FDMA combination. Each node maintains a TDMA-like frame, called super frame, where it schedules different time slots to communicate with its known neighbors by generating transmission/reception schedules during the connection phase. Each node either talks to one of its neighbors or sleeps in each time slot. The interference between adjacent links is avoided by assigning different channels to potentially interfering links with FDMA. Power saving is achieved using a random wake-up schedule, as illustrated in Figure 2.4., during the initial self-configuring phase, and by turning the radio off during the idling time. The following types of messages are exchanged between nodes (Nodes #1-3 in Figure 2.4) when searching for new neighbors,

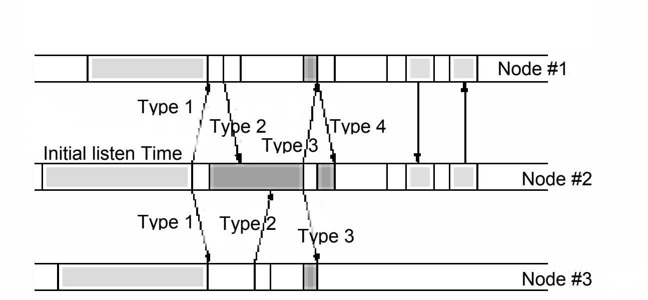


Figure 2.4: Node discovery phase during self-organization

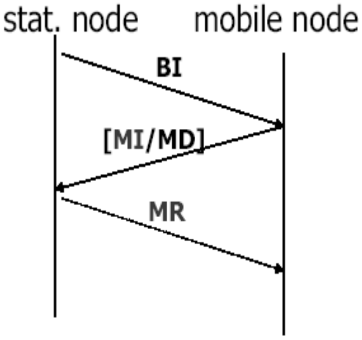
Type 1 – A short invitation containing a node’s ID and number of attached neighbors.
Type 2 – A response to Type 1 message. The node that sends Type 2 will be an invitee.
Type 3 – A response to Type 2 and this indicates which invitee was chosen
Type 4 – A response to Type 3 that contains invitee attached/non-attached and channel information.

As soon as a new link is discovered, the first time slot during which both nodes are free is assigned a channel and is added permanently to the probing node’s schedule. Each link operates at a different frequency to reduce collisions during channel assignment with other links. As time passes, each node updates and grows its neighbor list by attaching new nodes and eventually, most of the nodes in the WSN get recognized and self-configure with each other within themselves (it is possible that some of the nodes in the WSN never find a neighbor, and not attach to the WSN). When two or more nodes forming a subset of nodes, have coinciding super frames they need to be time synchronized. The transmission/reception pattern repeats periodically, once in every ‘T’ frames. This periodic pattern repetition is fixed for all nodes and is a local attribute of the super frame. After a link is formed, a node knows when to turn on its transceiver ahead of time for communication. This leads to substantial savings in energy.

2. The **EAR** (Eavesdrop- And-Register) algorithm [15] also looks at achieving power conservation based on TDMA-FDMA combination by enabling seamless connection of mobile nodes in the network. The drawback of this algorithm is the requirement of extra hardware and abundant bandwidth so that the nodes can tune the carrier frequency to be used in a link by randomly choosing from a large pool of available bands. The EAR algorithm has been designed for continuous communication between mobile and stationary nodes on the ground. It is desirable that the connection be setup with as few message exchanges as possible. Hence to conserve energy the mobile node assumes full responsibility for the connection setup, to process and decide on dropping connections. This is done by maintaining a registry at the mobile, stationary nodes.

   The algorithm makes use of four primary messages [15]:

   Broadcast Invite (BI) - The stationary node invites other nodes to join.
   Mobile Invite (MI) - The mobile node responds to the BI message.
   Mobile Response (MR) - The stationary node accepts the MI request.
   Mobile Disconnect (MD) - The mobile node informs the stationary node of a disconnect operation.



Moreover, two nodes may not be able to connect to each other if the unassigned regions in their schedules do not have enough overlap. In addition, like the previous energy conserving schemes, it does not take into account possible real-time requirements of the network, fairness and congestion control.

2.5 Qualitative comparison of Proposed MAC Schemes

Among the MAC protocols for WSNs discussed above, SMACS and EAR use a fixed allocation of duplex time slots at a fixed frequency while CSMA and the S-MAC protocols use a contention based access mode. To achieve maximum energy efficiency, the radios have a constant listening time in CSMA based protocols, while S-MAC uses periodic listen and sleep, avoiding idle listening. An interesting property of S-MAC for power saving is the ability to trade-off between energy and latency according to the traffic load conditions, and application-specific sensing. SMACS and EAR however require the radio to remain on during the entire set-up process, and be turned off during idle mode. TDMA-based protocols would generally have an advantage of energy conservation over contention based protocols, in that the duty cycle of the radio is reduced without any contention-introduced overhead and collisions. However, TDMA-based protocols need to form clusters and due to the frequently changing topology of WSNs, TDMA based attributes such as frame length, time slot assignment cannot adapt to the dynamic changes. In regard to scalability of the WSN, S-MAC would prove to be the most effective protocol. SMACS and EAR exploit the fact that available bandwidths can be much higher than the maximum data rate for sensor nodes. CSMA based MAC uses the application specific phase shift and pretransmit delay as the sensor network specific while S-MAC uses in-channel signaling with in-network processing. Table 2 below, provides a quick comparison of proposed MAC protocols for WSNs.

Table 2: Comparative evaluation of MAC schemes suggested for WSN

Attribute	SMACS, EAR	S-MAC	Hybrid TDMA/FDMA based	CSMA based
Channel Access Used	Dedicated assignment with fixed allocation of duplex time slots	Contention-based access	Centralized frequency and time division	Contention-based random access
Application Domain	For continuous traffic providing a bounded delay	Applications involving bursty traffic that can tolerate latency	Applications involving strict data latency requirements	Suitable for bursty traffic
WSN specifics	Exploits large available bandwidth in comparison to sensor data rate	Tradeoff between energy and latency	Fixed number of channels optimized for minimum system energy	Application phase shift and pre-transmit delay
Disadvantages	Appl. to static networks, Inefficient for applications having bursty traffic (inherent to WSN)	Reduced per-node fairness and latency	Close proximity of nodes to a high-powered base station	Cannot be used for delay-sensitive traffic

3. Proposed Scheme

WSNs are generally deployed for specific applications, where the life of the WSN depends on the resource-constrained environments of operation. However, for deploying robust WSNs extensively – in large numbers of unmanned battery-operated networked nodes functioning over a lifespan of a few years – there is a need for methods and protocols that promise better reliability and robustness for fault-tolerant deployment of WSNs. Since each wireless sensor node is equipped with a limited power reserve [8], care should be taken to make the most of the expendable power source for the node. Energy aware protocols and improved energy saving procedures at various layers of sensor networks have been a hot topic of research in the realm of WSNs. While many methods (refer Chapter 2) have been derived from traditional wired and wireless networks and suggested for WSNs, there is no unanimous choice on a universal MAC layer protocol yet.

In this chapter, we discuss our general purpose - application independent, CSMA based medium access control (MAC) scheme for energy efficiency in WSNs. Our MAC design makes use of a self-adaptive scheme based on an algorithm that adapts according to varying traffics in the WSN. We also envision sensor networks to be consisting of several clusters that are dynamically formed. WSNs are desired to be massively scalable and a cluster based medium access scheme can naturally lend itself towards minimizing the wastage of energy by reselection and reconfiguration process. We suggest that a collaborative approach by formation of dynamic clusters functioning with a traffic aware MAC design can influence the design of an energy efficient mechanism for application independent based sensor networks. In our proposed scheme outlined in this chapter we hope to contribute towards better energy efficiency in WSNs by suggesting a traffic-aware cross layer design for WSNs.

Our approach looks at increasing the energy efficiency globally using a self-learning, traffic adaptive scheme that renders the WSN more scalable and robust to varying topologies and application scenarios. We highlight the interesting features of our approach as follows,

(a) Scalability – our dynamic cluster based approach lends itself well to any scaling.
(b) Self-learning and adaptive to traffic intensities which ought to be an essential feature of any energy conserving scheme or protocol.
(c) Focuses on network wide energy savings thereby prolonging the network longevity rather than looking at a per-node basis.

Our scheme allows new/mobile/hibernating nodes to be discovered through the dynamic clustering approach. Due to the adaptive learning process of nodes capable of in-network data processing in the WSN, latency can also be reduced. While our method brings together methods on distributed and collaborative mechanisms of saving energy through dynamic clustering and adaptation to varying traffic conditions, the energy saved through this collaborative effort can pave the path for extensive WSN use and deployment.

3.1 General Outline

3.1.1 Problem Statement

From the extensive background study that we have presented in Chapter 2, we can note that considerable research is being done in the area of energy saving techniques for WSNs – with specific focus on energy awareness at different network layers (Figure 2.1) in WSNs. Furthermore, most of the proposed techniques are based on protocols for mobile ad hoc networks that do not look into the possibility of a cross-layer design strategy that can exploit the unique features of WSNs. We thus make the need for a universal protocol that can be applied to WSNs in general, our focus in this thesis. Specifically as a part of the study presented in the thesis, we focus such a research on optimizing the energy consumption by suggesting novel cross-layer architecture at the network/data-link layer for sensor networks. We have developed a scheme for better and improved energy efficiency in WSNs by combining the ideas of energy-efficient cluster formation [13, 14] and medium access together. The scheme is based on a collaborative approach supported by formation of dynamic clusters functioning with a traffic aware MAC (medium access control) scheme. Our MAC scheme incorporates a self-learning, traffic adaptive algorithm for varying traffic conditions inherent to the WSNs. To summarize, our cross-layer design for WSNs allows:

- nodes in the network to self-configure enabling ease of deployment,
- supports very high scalability of sensor nodes,
- higher throughput without much degradation of quality of sensed data, and
- improved energy-efficiency by adapting to varying traffic conditions inherent to WSNs.

3.1.2 Assumptions

We first summarize our assumptions with respect to the nodes in the WSN, which are used in the context of the proposed scheme. Our assumptions are listed below:

a. As WSNs comprise many miniature nodes deployed randomly in an ad hoc way, we assume that no greater than a few nodes can be a part of the dynamic cluster at any given time.
b. We also assume that during the self-configuration of nodes within each cluster, at least two nodes that are separated by a one-hop distance are awake during startup.
c. Communication within the cluster takes place over one-hop distance while traffic moves through the network over multi-hops [25, 26] to points that are connected to a much larger infrastructure [7].
d. Due to observed sensor data properties the required bandwidth is low.
e. The sensor nodes synchronize themselves according to a time synchronization scheme [19, 20].
f. All sensor nodes co-operate and collaborate for the same common task by sensing data at a fixed data rate.

3.1.3 Definitions

When a WSN is deployed (for example as shown in Figure 3.1 below) all the nodes within the WSN are assumed to have similar battery/power conditions. Each node is capable of a simple broadcast scheme using its radio antenna during the start-up (self-configuration) and initialization phase to recognize respective neighbors and start forming *clusters* [5]. We define the following terms,

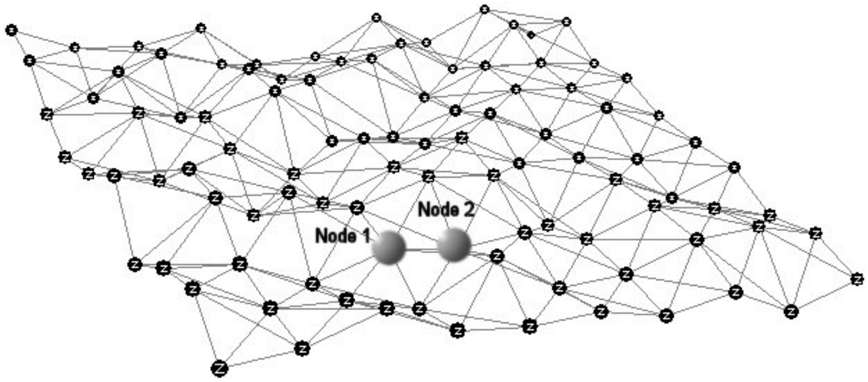


Figure 3.1: Section of a possible WSN showing 100 nodes deployed randomly (simulation model generated from SensorsimII [44])

Cluster - A cluster is defined as a collection of nodes that are located within one-hop (assumption c.) distance collaborating for the same communication task. Nodes within the cluster initially contend for the medium to become a *cluster-head*.

Cluster-head - The cluster-head defines the transmission area of the cluster and is responsible for the centralized control within the cluster. A cluster-head's radio remains "on" for the entire period of time before neighboring nodes re-contend for the position of a cluster-head. We discuss the cluster-head selection scheme in the following sub-sections.

Each sensor node becomes a part of a cluster by choosing the cluster-head that is "closest" to itself in a signal strength sense rather than in spatial sense (distance). The choice of signal strength over distance is justified as topographic deployment of WSNs can cause physical obstructions (affecting line-of-sight [29]) between various nodes. Furthermore, due to energy constraints (continuous consumption in sensing, data-processing, communications) nodes that are closer in distance can have lower energy levels than farther nodes and run out of battery power quickly. In conjunction with signal strength we define an optimum threshold *battery level* ($\mathbf{T_{bl}}$ discussed in section 3.2.2) for each node. Individual nodes in the close vicinity of other single-hop, 2-hop nodes contend for the cluster-head position only if their individual battery level is in excess of T_{bl}.

Once the nodes have self-configured and formed clusters they adjust their sleeping times adaptively using the algorithm outlined in Section 3.4. We illustrate our detailed approach in the following sub-sections.

3.2 Dynamic Cluster communications

3.2.1 Initialization Phase

The method of clustering and self-organization is explained with the help of the scenario shown in Figure 3.2. Nodes 1-9 (represented as N1-N9) shown in the figure are assumed to be the part of an extensive WSN. Nodes numbered N1-N9 need to self-configure and form clusters. The first node to broadcast in the vicinity of its neighbors [38] is chosen as the first cluster-head by nodes that are one-hop away from the broadcasting node. By the use of a random timer included as a part of the simulation process we ensure that no 2 nodes in the vicinity of each other can broadcast at the same time.

The broadcast mechanism by a node is done to make its presence known to all neighbors at single-hop distance. Based on the assumption that at least one node is awake at one-hop distance, the corresponding cluster-head sets a timer for which it decides to stay as the cluster-head. The time to act as a cluster-head can be preset and incorporated into the firmware of each sensor node prior to deployment for node-level fairness, so that any given node can serve as a cluster-head. After the cluster-head sets the timer on, it periodically advertises its presence to neighbors at a single hop distance. For example, N3 broadcasts to N2, N4 N6, and N7 in Figure 3.2 below.

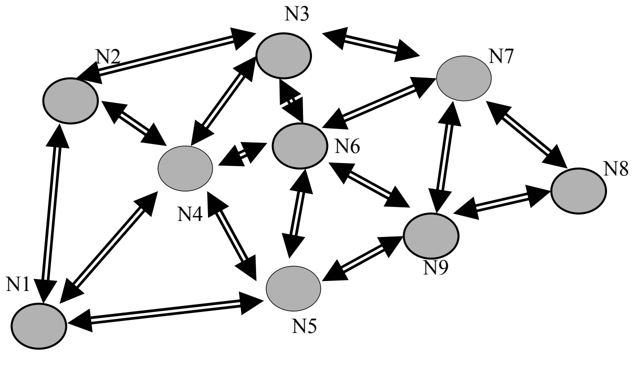


Figure 3.2: A group of nodes self-configuring and forming clusters

During the periodic broadcast by the cluster-head various nodes at one-hop distance can join the cluster based on the signal strength of the received signal. A handshake takes place between the broadcasting cluster-head and the non-cluster-head neighbors, before any data transmission can begin. The node that communicates with the cluster-head acknowledges the broadcast control packet transmitted by the cluster-head. The node replies back to the cluster-head in a procedure similar to the LEACH [17] protocol, by sending a cluster-membership request message acknowledging the broadcast packet from the cluster-head. The message is a short message that contains the node's and cluster-head's ID using a non-persistent CSMA MAC scheme [27]. These initial control messages are short in length and hence a higher transmit power for these control packets is the suggested choice to keep the latency low and overcome the hidden terminal problem. The hidden-terminal problem may arise if the transmit power for the control packets is low, because nodes close to the cluster-head acknowledging the broadcast packet with lower transmit power might allow the remaining nodes near the cluster-head to not sense the on-going and transmission and these other nodes might themselves start transmitting acknowledgment packets concurrently causing heavy energy wastage due to collisions, in the initialization phase itself. Furthermore, by using higher transmit power for the control messages/packets latency can be reduced implying more sleeping time for the nodes after initialization. A flowchart showing the initial cluster formation and initialization is shown in Figure 3.3.

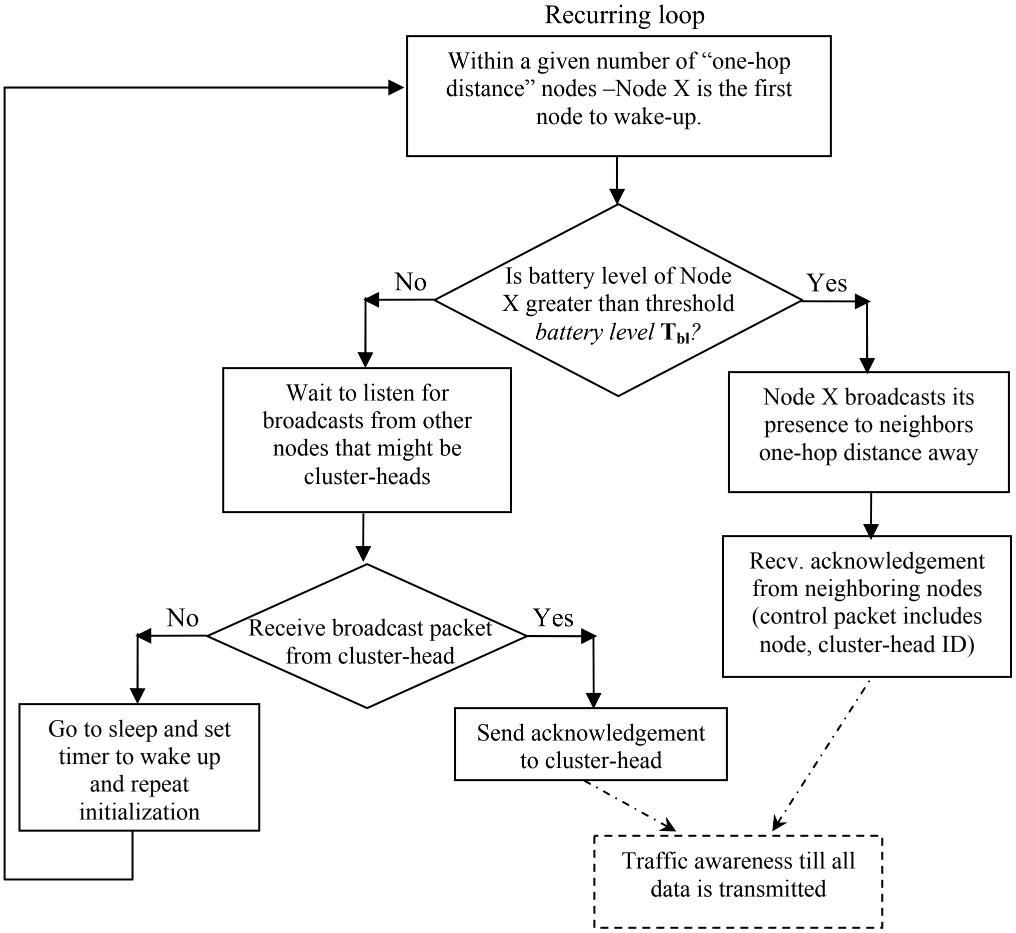


Figure 3.3: Flow-chart for initial cluster formation and initialization

We illustrate the discussion above with an example. In figure 3.2, N4 first broadcasts its presence. Let's say N1 (cluster neighbor #1) acknowledges the presence of N4 during N1's initial startup time. This makes (N1, N4) pair a part of the initial cluster. Over periodic broadcasts from N4 advertising the cluster-head ownership, let us say N2 also wakes up and acknowledges the membership of the cluster that N4 is the cluster-head. Similarly N3 joins the cluster making the cluster 4-node strong while data-transmission is in progress in-between periodic broadcasts from N4. Even though N6 is located at one-hop distance away from N4, it joins the advertising N5 instead of N4 based on the advertisement and signal strength of N5. Following a similar methodology N7, N8, N9 form a cluster simultaneously as various other nodes work towards forming their own clusters. As the clusters are being formed we assume that an underlying periodic time synchronization scheme [10, 11] as part of the control packet handshake can be used for network wide synchronization to make the WSN robust towards synchronization errors. While the nodes are self-configuring, they also become aware of the signal strengths and amount of time each node is ready to be the cluster-head (as part of the broadcast packets).

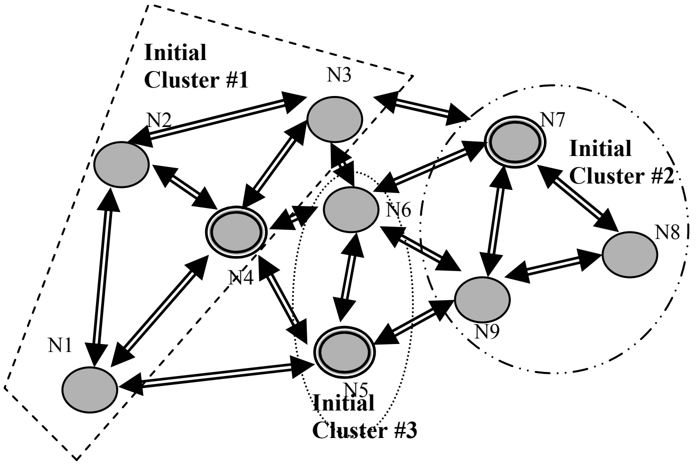


Figure 3.4: Group of nodes from Fig 3.2 self-configuring to form clusters

A completed picture of the self-configuration phase of the scenario is shown in Figure 3.4 with the double circled nodes representing the cluster-heads and the dotted periphery encompassing the clusters.

3.2.2 Data Transfer Phase and Reconfiguration Phase

In the previous sub-section we discussed how nodes self-configure and communicate with an exchange of their ID's and the cluster-head ID. Based on our assumption that nodes always need to send some data to the cluster-head after self-configuration, there is an exchange of Ready To Send/Clear To Send (RTS/CTS) packets between the cluster-head and transmitting node [27]. The cluster-head must keep its radio receiver on to receive all the data from the nodes in the cluster. After all the data packets exchange has been completed within a cluster, each node by itself can decide if it wants to contend for a cluster-head position after the clusters disintegrate.

The aforementioned choice of reconfiguration, safeguards any single node (that acts as the cluster-head and keeps its radio on for all the time that it functions as a cluster-head) from being drained out of its battery-power by repeating the process periodically over a time T_f. T_f is an estimated period after which the clusters disintegrate (in the sense that they re-search for their neighbors) to form a new cluster headed by a new cluster-head. T_f depends on several parameters such as application-requirements; network topology and mean lifetime that WSN needs to operate.

Another factor to decide the contention for the cluster-head position after T_f is the optimum threshold *battery level* (T_{bl}) – that can change over a period of time (6-7 months or based on the application-specific deployment). This can be achieved by updating the firmware of the sensor node that periodically reduces T_{bl} prior to deployment. T_{bl} would generally reduce over time as any random sensor node in the WSN would have depleted its power reserve at least to a small degree in that time (based on the static and dynamic energy consumption). Thus, by allowing optimal self-configuration of the sensor nodes to automatically get divided into several dynamic clusters, different clusters can disintegrate according to a repetitive T_f schedule and re-cluster newly. After time T_f, the initial cluster scenario changes to the one as shown in Figure 3.5. Every time the nodes in the network reconfigure – they search for neighbors and hence newly introduced/ sleeping nodes that wake-up after extended periods of inactivity join the network over time and get discovered. This makes our approach much more scalable in comparison to existing procedures.

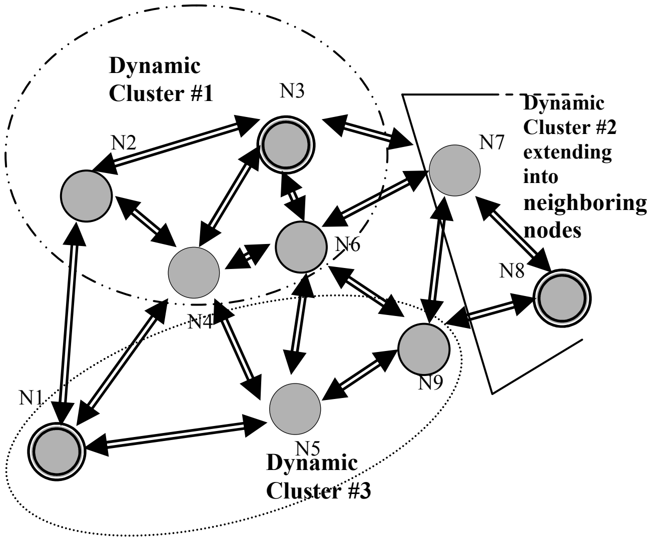


Figure 3.5: Group of nodes from Fig 3.2 that have self-configured and formed clusters

The flow-chart for the data transfer and reconfiguration phase is shown in Figure 3.6.

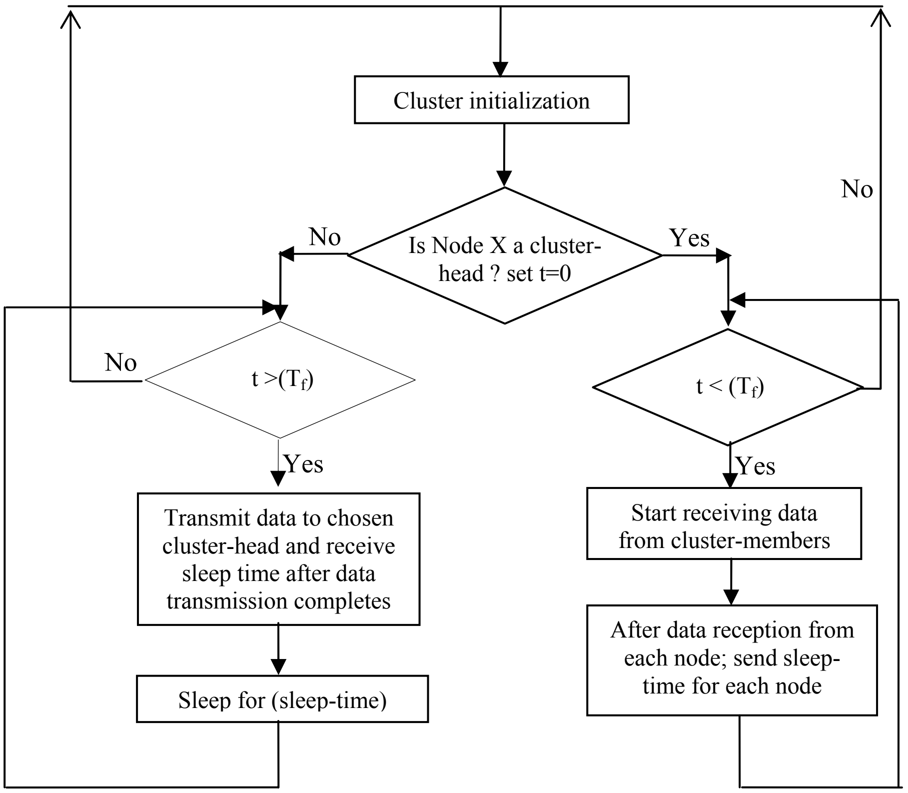


Figure 3.6: Flow-chart for data packet transfer between nodes and cluster-head

3.3 Adaptive scheme for traffic adaptability

The sensor node architecture shown in Figure 1.3 has an onboard radio that operates in four modes of operation, namely *Transmit*, *Receive, Idle,* and *Sleep* modes. Transitioning between these distinct modes involves valuable power dissipation attributed to the transient activity in the radio hardware. However, if we can adapt the sensor radio listening, idle and sleeping times according to a duty cycle determined by a repetitive self-learning scheme enormous amounts of power can be saved globally in each cluster over the entire WSN. Our focus of this learning scheme is essentially centered around learning from the variable and highly correlated traffic inherent to WSN's (owing to event-driven characteristics that cause very low activity over extended periods of time, with sporadic outbursts of very high traffic).

For proper and reliable data-sensing between various clusters in the WSN, each node maintains a schedule table that stores the cluster-head information and the schedules of all the in-cluster neighbors based on the sleep-times received from the cluster-head. Every node goes through the following steps to store the schedules of all its known neighbors.

1. The node scans the channel waiting to listen from the cluster-head for a fixed amount of time (T_{on}).

2. If the node receives a schedule from a cluster-head during T_{on} it follows the schedule by updating its schedule table.
3. If the node does not receive a schedule during T_{on}, it goes off to sleep for time T_{off} and then goes back to the 1st step after waking up.

For the purpose of transfer of messages between cluster-members and cluster-head, the messages are divided into many short independent packets (typically 20-30 Bytes long). RTS and CTS packets are then used in contention for each independent packet. The long message that has been fragmented into the small sized packets is transmit in bursts till the complete message is transmitted to the cluster head. Only one RTS packet and one CTS packet is used per packet. The timing scheme between different cluster members is shown in Figure 3.7 below.

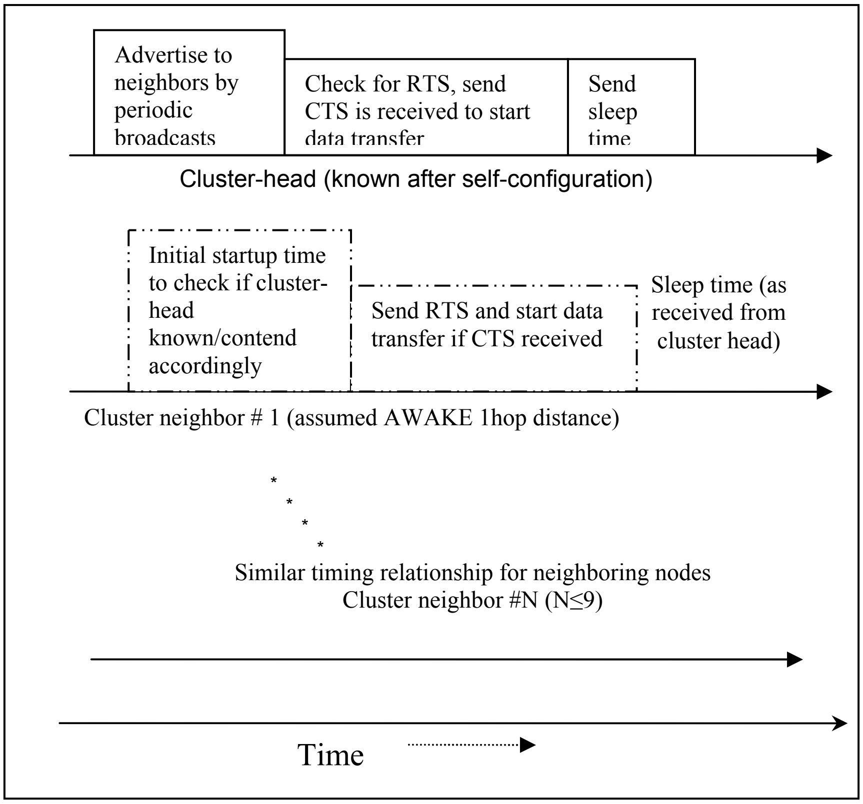


Figure 3.7: Timing scheme followed within the cluster

As discussed in the previous sub-sections the cluster-head first broadcasts its presence to the neighboring one-hop away nodes and after the self-configuration phase every time a data fragment is transmitted, the transmitting cluster-member node waits for an ACK from the cluster-head (followed by the sleep-time if the complete message is transmitted). If it fails to receive the ACK, it will re-transmit the non-ACKed fragment immediately. Each packet is timed for the amount of maximum time that it can take to be transmitted. If a neighboring node hears a RTS or CTS packet, it will go to sleep for the time that is needed to transmit all the fragments. This is done so that the node that does not have to send packets immediately can go off to sleep and save energy. Unlike the 802.11, where the RTS and CTS only reserves the medium for the first data fragment and the first ACK, our scheme sends an ACK after every packet transfer. Furthermore, through the transmission of ACK after every data packet the purpose of using ACK after each data fragment is to check for reliable transfer of the packet and also to prevent the hidden terminal problem. The algorithm for adaptively making the nodes within the cluster go to sleep for greater times based on the varying taffics in the WSN is shown in Figure 3.10.

```
Loop
 pattern_type learn_traffic_pattern

{Repeat for-loop till time to relinquish control as
          cluster-head node

   for (each packet CTS sent by cluster-head node)
     {Non-clusterhead node sends RTS to clusterhead;
      Wait for cluster-head to send CTS;
      Begin transfer of data by setting parameter
          $T_{adapt}$;
      Data transfer over stop $T_{adapt}$;
      Non-cluster-head node receives sleeping time
        at end of data transfer from cluster head;
      Exit loop;
     }

Cluster-head calculate average sleeping time sent to
      transmitting nodes, average $T_{adapt}$ iteratively;

Cluster-head buffers collected data from nodes,
      $T_{adapt}$, sleep times to send to new cluster-head;

} //Repeat loop every time cluster changes
```

Figure 3.8: Outline of adaptive algorithm

The algorithm utilize a random timing scheme based on the T_{adapt}, for varying time-intervals by averaging the sleeping times that the cluster-head issues to the neighboring nodes within the cluster. The algorithm optimally uses the T_{adapt}, sleeping times based on the amount of traffic in the network (based on the amount of data it receives during T_f) and allows switching between on/sleep times of the radio by adapting to varying traffic loads. Energy is thus saved locally per cluster, as the nodes within the cluster have greater sleeping time as non-cluster-head nodes can either follow the optimized sleeping time sent from the cluster-head, or go to sleep if no schedule is received from the cluster-head after start-up time. Since the cluster is moving dynamically over the WSN after T_f the local energy saving extrapolates to a global energy saving. Furthermore, for greater reliability and robustness, at the end of T_f – the data collected by the cluster-head is buffered and sent to the next cluster-head towards points that are connected to a much larger infrastructure for data aggregation [7,10].

4. Design models and MAC description

We have chosen the ns-2 simulator [39] for this research because it realistically models arbitrary node mobility as well as physical radio propagation effects such as signal strength, interference, capture effect, and wireless propagation delay. The simulator also includes an accurate model of the IEEE 802.11 Distributed Coordination Function (DCF) wireless MAC protocol.

4.1 Sensor Node design in ns-2

To implement our protocols, we added several features to ns-2 [39], an event-driven network simulator with extensive support for simulation of wireless network protocols. Developed at the University of California at Berkeley and the Lawrence-Berkeley National Laboratories in collaboration with the VINT (Virtual InterNetwork Testbed) project, ns-2 has a simulation engine written in C++ with a command and configuration interface using OTcl. Network topologies can be easily described using the primitives Nodes, Links, Agents, and Applications, where Nodes represent end-hosts in the network, Links are the connectors through which Nodes communicate, Agents are used to implement different network protocols and are the points where packets are created and consumed, and Applications are used to generate data and perform different application-specific functions. Once the topology has been created, simulations can be run by starting the Applications on different nodes at various points in time. While ns-2 was developed as a simulator for wired networks, researchers at Carnegie Mellon University added extensive support for wireless networks. The CMU additions include mobile nodes, MAC protocols, and channel propagation models. Figure 4.1 shows our implementation of a sensor node. It has been derived out of the ns-2 specifications [39] for the design of a

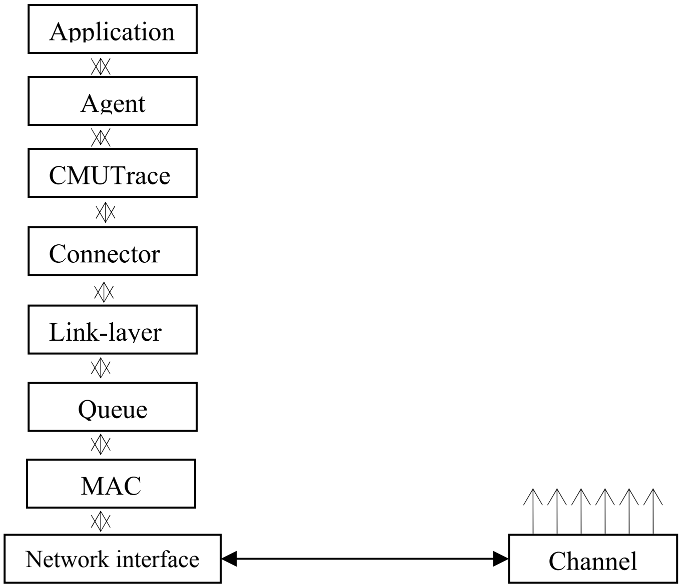


Figure 4.1: Sensor node implementation in ns-2 (derived from mobile node model in ns-2 [39])

mobile node. The Application class is written using the Tcl front-end, while the other functions that make up the node are written using the C++ engine. The Application creates “data packets" that are sent to the Agent that performs the transport and network-layer functions of the protocol stack. The Agent sends packets of data to CMUTrace, which writes statistics about the packets to trace files. The packets are then sent to a Connector which passes them to the Link-Layer for data-link processing. After a small delay, the packets are sent from the Link-Layer to the Queue, where they are queued if there are packets ahead that have not yet been transmitted. Once a packet is removed from the Queue, it is sent to the MAC, where media access protocols are run. Finally, the packet is sent to the Network Interface, where the correct transmit power is added to the packet and it is sent through the Channel. The Channel sends a copy of the packet to each node connected to the channel. The packets are received by each node's Network Interface and then passed up through the MAC, Link-Layer, Connector, CMUTrace, and Agent functions. The Agent de-packetizes the data and sends notification of packet arrival to the Application.

4.2 Incorporated MAC functionality

We have incorporated our MAC protocol type into the available functionality of S-MAC in ns2. When a packet is used for an advertisement or a join-request message, and/or a data message being sent to the cluster-head it is transmitted using a CSMA approach. The MAC protocol is always chosen such that it reduces energy dissipation by allowing nodes to remain in the sleep state for as long as possible (using our adaptive algorithm outlined in Figure 3.7) and minimizing collisions (using CSMA to reduce the number of collisions). Non-persistent CSMA is used for the experiments. To perform CSMA, the node senses the channel before transmission. If the channel is currently being used by someone else, the node sets a back-off timer to expire after a random amount of time, where the timer is chosen uniformly with a maximum time equal to the transmit time of the packet it is waiting to transmit. This back-off policy for CSMA is effective because all nodes are transmitting packets with the same length during a given time. Therefore, the maximum amount of time that the channel will be busy is equal to the amount of time it would take to transmit the node's packet. Once the back-off timer expires, the node again senses the channel. If it is still busy (presumably someone else captured the channel first), the node again sets a back-off timer. This continues until the node senses a free channel. Once the channel is free, the TX STATE variable is set to MAC SEND and the node passes the packet onto the NetworkInterface. The node must also set a transmit-timer so it knows when it has finished transmitting the packet (and can reset the TX STATE variable to MAC IDLE). This is important because a node cannot transmit two packets at the same time, and a node cannot receive a packet while it is transmitting. Since data is not actually transmitted in ns-2, the required measurement variables are just listed in the header of the packet. The following pseudo-code outlined in Figure 4.2 derived from [28] performs these functions:

```
SensorMAC::send(packet)
{
if NodeTransmitting || NodeReceiving || channelBusy
then
time = random_number(0,TX_Time(packet))
backOffTimer->start(packet, time)
return
TX_STATE = MAC_SEND
TXTimer->start(TX_Time(packet))
networkInterface->recv(packet)
}
SensorMACbackOffTimer::finish(packet)
{
send(packet)
}

SensorMACTXTimer::finish()
{
TX_STATE = MAC_IDLE
}
```

Figure 4.2: Pseudo code

As long as there are fewer number of transmissions heard at the receiving node, it is assumed that the reception of the packet is successful. A node cannot receive a packet while transmitting. If the node is transmitting as a packet arrives, the received packet is marked as erroneous. However, it is not dropped because the receiver may be busy receiving this erroneous packet after the transmitter has finished. All packets that contain errors are dropped in the link-layer of the stack. If the node is currently receiving another packet (pktRx) when this new packet (p) arrives, two situations can occur.

a. First, pktRx might be sent with enough power to swamp out the reception of the new packet, p. In this case, capture occurs and p is dropped.
b. On the other hand, if pktRx does not have enough power to swamp out p, both packets collide. In this case, the packet that will last the longest for reception is kept and marked as erroneous and the other packet is dropped.

By keeping the packet that will last the longest, the receiver is busy for the maximum amount of time. Again, the packet that is kept will be dropped in the link-layer of the stack. If the node is neither transmitting nor receiving when the new packet arrives, the RX STATE variable is set to MAC RECV and a timer is set that expires after the length of time required to receive the packet. When the timer expires, the node checks the address field of the packet. If the address is the node's address (or the broadcast address), the packet is sent up the stack to the queue. Otherwise, the packet is not intended for this node and is dropped.

5. Discussion of Results

This chapter presents the performance analysis of the energy aware MAC Protocol for WSNs by incorporating the proposed method of adaptability to traffic. The procedures discussed in Chapters 3, 4 have been analyzed and validated under various scenarios with different configuration parameters. The node and radio characteristics have been referenced from published sources and data sheets [Appendix IV].

The traffic inherent to WSNs is highly sporadic and does not necessarily follow any specific traffic pattern. We first analyze the effect of such varying traffics (attributed to the event-driven sensing operations) on the energy consumption in WSNs. A simple two-node scenario shown in Figure 5.1 below, is considered for the analysis.

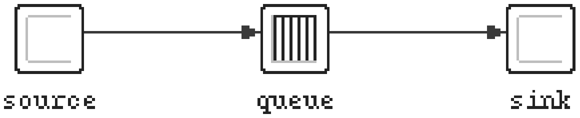


Figure 5.1: Simple two-node block diagram (where source is a transmitter and sink a receiver)

Messages are stored and forwarded for processing by a queue model within individual sensor nodes, as per requirement. It is assumed that a robust routing protocol [18] at the networking layer exists that can route and provide connections for routing packets between communicating nodes in the WSN.

Traffic and Topology model

The arriving traffic for a node is modeled using the single source producing packets at rate λ such that,

$$\lambda=\Sigma\lambda_i, \qquad \textbf{...(1)}$$

where, 'λ' is the arrival rate of the arriving traffic
'i' represents neighboring nodes in the WSN

The number of limited requests/traffic load that an individual node can process is varied by setting/changing the inter-arrival period of the arriving traffic (equation (1)) in the WSN. The queues are represented by a single block within the MAC interface, where temporary requests and packets are queued at each node. The queue size can be changed as per the application specifics. The spatial distribution of the sensor nodes is assumed to be uniform with possible mobile nodes (if any) traveling at very low speeds, thereby allowing mobility effects to be neglected. We thus emulate a very generic topology for WSNs. In a realistic WSN scenario, different regions within the topology will have different traffic rates. To account for this, we use the *exponential* distribution density function (reference APPENDIX II),

$$f(x) = ae^{-ax} \qquad \text{...(2)}$$

(where 'a' represents the mean of the distribution) and the *pareto* density function (reference APPENDIX III),

$$f(x) = ax^{-(a+1)} \qquad \text{...(3)}$$

to plot the normalized energy consumption vs. varying traffic between two nodes that are separated by a one-hop distance away (source and sink in Fig 1).

We define the following simulation parameters for the analysis of varying traffics with energy,

E_{IDLE} = Energy used in idle listening
E_{RX} = Energy required for receiving a transmit bit
E_{TX} = Energy required for transmitting a bit
T_{off} = Radio sleep time (no meaningful traffic sent/received)
T_{on} = Radio active "on" time for the receiving/transmitting bit

We define the following ratios,

$$T = \frac{T_{off}}{T_{on}} \qquad \text{...(4)}$$

$$E = E_{IDLE} : E_{RX} : E_{TX} \qquad \text{...(5)}$$

For the simulations, we set different values for equation (4) as observed by [1], [40] respectively. These values are,

$E = 1 : 1.05 : 1.4$ **...E1**
$E = 1 : 2 : 2.5$ **...E2**

For different traffic conditions as discussed in the various Phases (Phase 1, Phase 2, Phase 3) [1], we set the values for equation (3) as follows, with a fixed queue size of 5 packets. We choose a fixed queue size of 5 packets (with each packet = 2 bytes length) due to the limited processing capabilities and buffer sizes of hardware nodes [Appendix IV]. The analysis

Maximum Sleeping Time → $T = 9:1$ **...T1**
Moderate Sleeping Time → $T = 1:1$ **...T2**
Minimum Sleeping Time → $T = 1:9$ **...T3**

For the simulation purposes we use the parameters in Table 3 for the radio transmitter and receiver. In Figures 5.2-5 the normalized Energy consumption is plotted on the Y axis by varying it against the arriving traffic on the X axis [16].

Table 3. Parameters used for simulation (source data sheets [Appendix IV])

Parameter	Value
E_{IDLE} (Energy used in idle listening)	1 mW
E_{TX} (Energy required for transmitting a bit)	25.0 mW
E_{RX} (Energy required for receiving a transmit bit)	14 mW
E_{SLEEP} (Energy used in sleep mode of radio)	15 μW
Energy expended during Transition of state (Sleep-Receive)	40 mW
Energy expended during Transition of state (Transmit - sleep)	25 mW

Figure 5.2 is plotted using an *exponential* distribution, with the value of mean distribution 'a' = 0.5. For the case of (T1, E1) used in Figure 5.2, we observe that the T_{off} time is much larger than T_{on} according to the ratio of $T = 9:1$. This can be interpreted by the reader in either of the two ways suggested below,

- That a given node has a much longer sleeping time (in ratio of 9:1) w.r.t the radio's awake time, or,
- That in a section of the WSN, percentage of sleeping nodes is much larger than those that are awake (in proportion of 9:1 as suggested by T1).

Such a ratio is chosen to see the effect of varying traffic in a scenario where most of the nodes have maximum sleeping time.

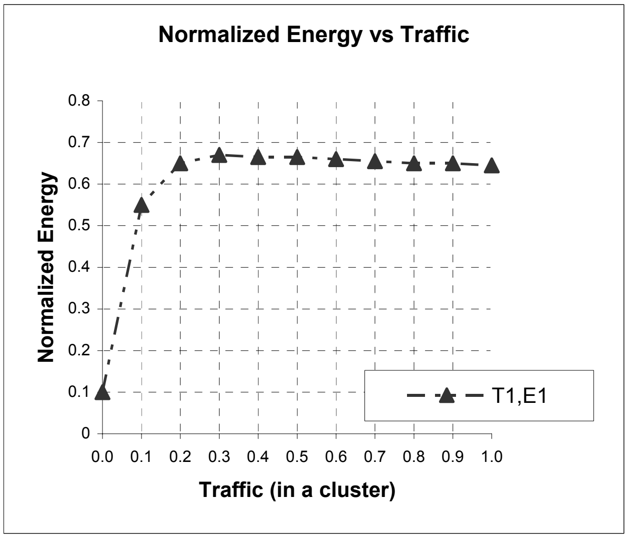


Figure 5.2: Normalized Energy vs. Traffic (Exponential distribution a=0.5)

The plot above shows that when the traffic is almost negligible, a higher sleeping time T_{off} results in very low energy consumption, as would be expected. However, when the traffic increases, a higher T_{off} does not necessarily keep the energy consumption low, as observed by the steep rise in the gradient. This can be attributed to the fact that an increase in traffic (meaning either an event has been detected or self-configuration is taking place) requires that more nodes be awake for data/packet transfer. However, due to the choice of T1, the nodes with their transceiver radios on (even though only a very small proportion) have to continually keep sensing the medium with repeated broadcasts of control packets waiting for nodes that are one-hop distance away to reply. The energy consumption thus increases rapidly to a maximum, and then reduces slightly.

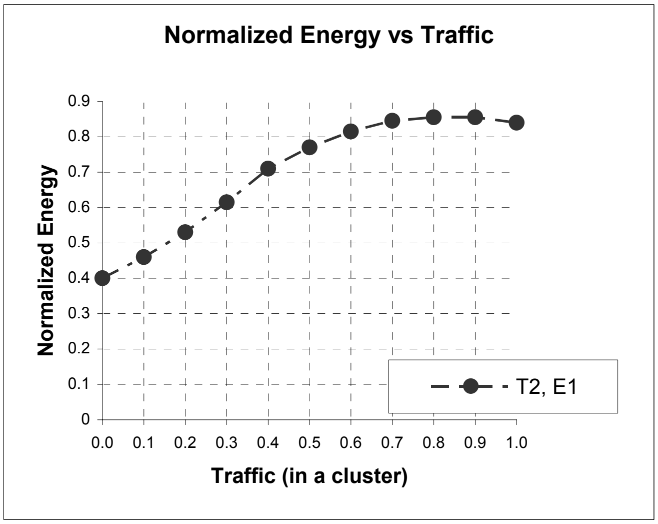


Figure 5.3: Normalized Energy vs. Traffic (Exponential distribution a=0.5)

Following a similar analogy for the case discussed in Figure 5.2, we choose the ratio of $T_{off} : T_{on} = 1$, for Figure 5.3 as is the case of (T2, E1). The choice of T_{off} and T_{on} times being in the same proportion that the probabilities of finding a node sleeping or awake during T2 is 0.5, i.e., the nodes have an equal probability of being found awake or asleep at any given instant during the medium traffic scenario. Another interpretation to the choice of T2 is that a given node is either awake or asleep for half of any given time. When the traffic is almost negligible, the energy consumption starts at 40% of the normalized energy level to increase gradually with increasing traffic. When self-configuration and/or event-detection takes place the control packets along with the data packets being exchanged between the transmitter and receiver nodes (source, sink) increases the energy consumption. As 50% of the time nodes are sleeping according to T2, the increasing number of packets with increasing traffic proportionately increases the energy consumption. This almost linear rise in the traffic-energy consumption can be observed in Figure 5.3. The trend in the increasing energy consumption due to increasing traffic can be attributed to the increasing number of packets that are being transmitted. This causes a reduction in the idle-listening time of the radio (Table 5.1), thereby increasing the energy consumption.

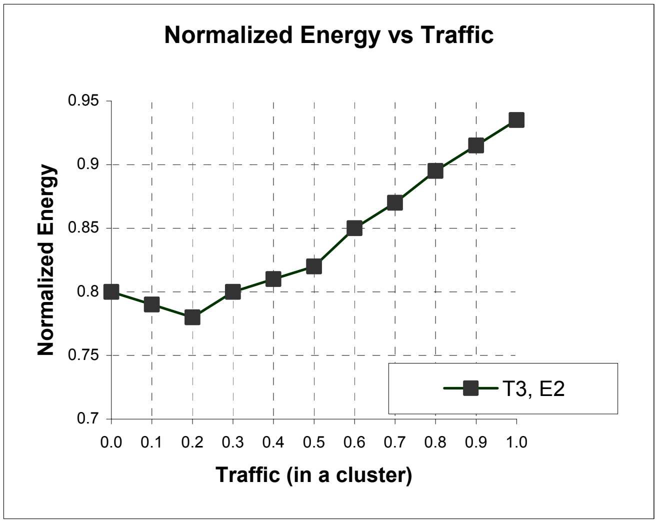


Figure 5.4: Normalized Energy vs. Traffic (Exponential distribution a=0.5)

In Figure 5.4, the normalized energy vs. traffic analysis is plotted for the minimum sleeping time case of (T3, E2). The nodes are awake for 90% of the time and are continually listening (idle-listening) even when no data is transmitted. As expected the energy consumption is very high starting off at about 80 % of the maximum normalized energy even for negligible traffic. The energy consumption reduces to about 78 % of the maximum only to rise to almost 95 % with increasing traffic. This trend is observed due to reduction in the idle-listening time of the radio, and a cumulative increase in the E_{TX} and E_{RX} due to the increased traffic. Over-hearing during heavy traffic due to the high proportion of awake-nodes also increases the packet retransmissions thereby increasing energy consumption as observed by the rising gradient in the plot.

To qualitatively compare the different trends observed in Figures 5.2-4 above, we collectively plot the curves for different $T_{off} : T_{on}$ ratios (i.e. T1, T2, and T3) in Figure 5.5. The plot is used for the analysis of the varying sleep times on varying traffic conditions in the source-sink model discussed in Figure 5.1. For almost negligible traffic we can see that in the case of T1 minimum energy consumption takes place. The energy consumption in the case of T2 is higher than in the case of T1 but lower in comparison to T3. More energy is expended by a given node when its radio is in the E_{TX} and E_{RX} mode, than when it is in the E_{IDLE} mode. Hence during the low traffic phase (lower bound on normalized traffic – 10%, upper bound on normalized traffic – 40%) we observe that energy consumption increases for all the T1, T2, T3 traffics. The energy consumption for T3 overshoots T1, T2 by a large margin (15-30% of maximum normalized energy) due to the high awake times of the nodes. One interesting observation is that though we would expect the energy consumption for T2 to be higher than T1, it is not the case. This can be attributed to the fact that due to the increased sleeping time of the nodes in T1, more energy is expended when switching occurs from sleep-awake state. However, for heavy traffic more energy can be saved by putting the nodes to sleep. As the traffic increases, we observe that for (T2, E1) and (T3, E1) the energy consumption does not increase manifold in comparison to lower traffic conditions. A reduction in the idle-listening times in the WSN can be used to explain the cause. At medium traffic conditions, the energy consumption increases sharply for (T2, E1) and actually stays almost constant for (T1, E2).

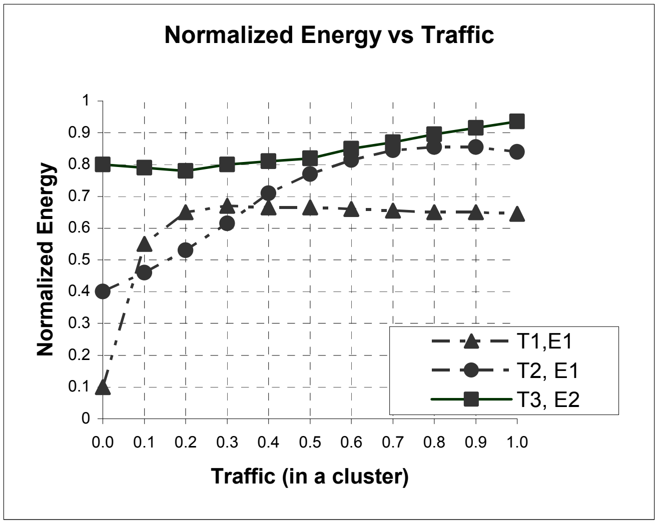


Figure 5.5: Normalized Energy vs. Traffic (Exponential distribution a=0.5)

Figure 5.6 is plotted with the same metrics as Figure 5.2-5, except that now we use the *pareto* distribution with the mean distribution of 'a' = 0.5. We observe that the trend followed by the curves in the case of exponential distribution remains the same for the pareto distribution as well.

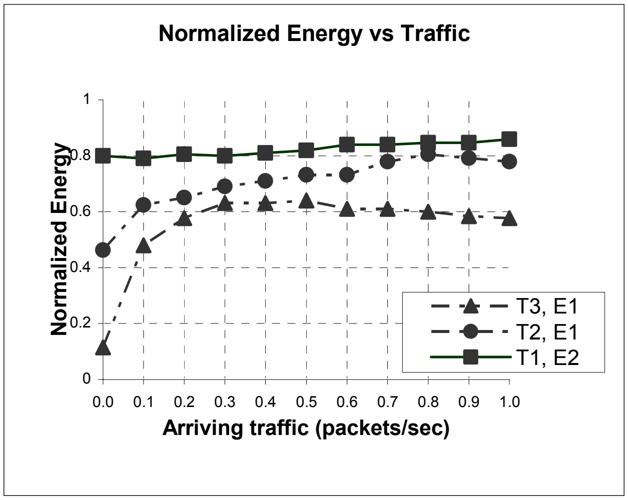


Figure 5.6: Normalized Energy vs. Traffic (Pareto distribution a=0.5)

From the analyses of Figures 5.2-6, we come to a conclusion that increasing the sleeping times of the nodes in a WSN does not always necessarily cause increased energy saving. For the case of the low traffic phase in the Figures 5.2-6, by increasing the sleep times of the nodes more energy can be saved. However, in the medium and heavy traffic portions of Figures 5.2-6, introducing some “traffic-adaptability” into the node’s communication architecture can allow improved energy saving. If a mechanism of choosing the optimum sleeping time is incorporated into the medium access control of the node, we should be able to strike a trade-off between the sleeping time and energy consumption characteristics, hoping to reduce energy consumption and save greater amount of energy per bit of useful transmission.

In the results that are plotted in the subsequent sections, the adaptive algorithm outlined in Chapter 3 has been incorporated as a part of the MAC layer functionality used in the operation of the simulated sensor nodes.

We choose the following sensor network settings for our simulations. Sensor nodes are deployed over a terrain of 100 m x 100 m. The power of the sensor radio transmitter is set so that any node within a 10 meter radius is within communication range and is called a neighbor of the cluster-head. The radio used in the nodes can communicate within a circular radius of 10 m (radio range), has data rate of 10 kbps and employs CSMA type modified MAC protocol. The packet format used by the radio consists of 15 bytes of data and 2 bytes of header. Thus, each packet takes around 13.6 ms for transmission. Nodes are located randomly in the sensor terrain. Note that data is generated only when an event happens, and we assume that nodes have already detected and have data to transmit. The processing delay for transmitting a message is randomly chosen between 5 ms and 10 ms using the functionality of ns-2 [39]. The node operates at 25 mW, 14 mW and 0.015 mW in transmit, receive and sleep state respectively, same as in [Table 3]. Moreover, as is typical of short range radios, the power during transmit, receive, and idle (passive listening) modes are comparable, and for purposes of illustration we consider them to be equal. The initial energy of the nodes (fixed by the battery used in the nodes) is 200J.

5.1 Metric Analyses

In this section, we use simulation setting discussed above to study the performance of the adaptive MAC layer protocol. We use the following performance metrics:

- Throughput: the rate of data packets arrived to the cluster-head.
- Packet drop count: the number of dropped data packets due to buffer overflow.
- Average energy consumed per packet: the average energy consumed in transmitting and receiving a data packet.

5.1.1 Throughput curves

With each node in the cluster sending data at a rate of 5 packets/second, Figure 5.7 shows the resulting aggregate throughput. We observe that with the increasing cluster size from 3 member nodes to 8 member nodes the throughput in the cluster raises almost very gradually. In the case of RTS/CTS mechanism used by the IEEE 802.11 MAC without fragmentation, which treats each data packet as an whole independent packet and uses RTS/CTS for each of them – this MAC gives the lowest throughput. 802.11 has no

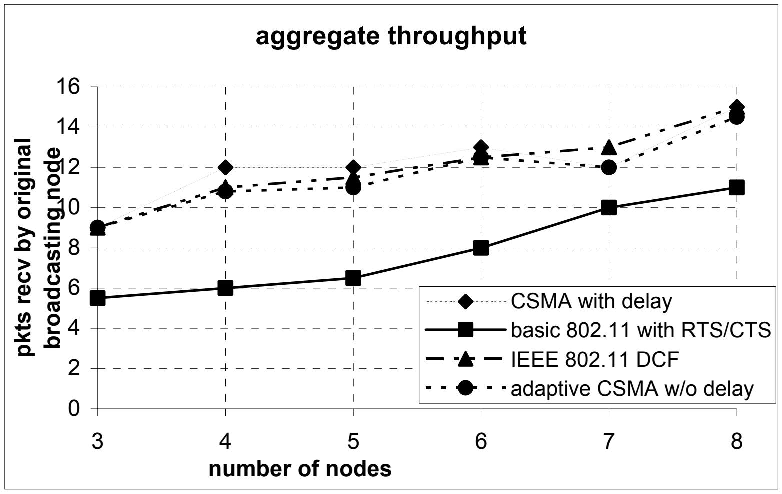


Figure 5.7: Throughput Analysis (buffer size = 5 bytes)

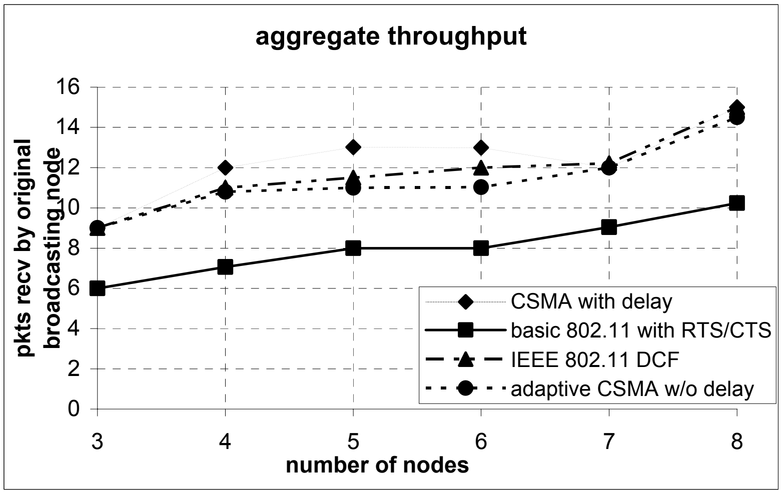


Figure 5.8: Throughput Analysis (buffer size = 8 bytes)

random delay and has a constant listen period, where the ACKs (acknowledgments of successful data received) provide collision detection and trigger a backoff mechanism. All the variants of our schemes achieve greater bandwidth than the 802.11 scheme with its explicit ACKs. In the case of the 802.11 Distributed Co-ordination Function (DCF) we observe that the throughput is better than the regular 802.11 case. The improvement of throughput for the case of DCF is observed as overhearing is reduced.

The schemes with constant listen period and no random delay achieve highest bandwidth, especially where the load is just below the channel capacity (3 to 4 nodes). However, their aggregate bandwidth is not very robust, as indicated by the two dips in the figure. The dips are caused by repeated collisions, which these schemes are incapable of eliminating. The remaining schemes, with random delay or random listening intervals achieve slightly less bandwidth, but are more robust. As network load exceeds the channel capacity, all schemes except 802.11 utilize about 75% channel capacity, or 15 packet/s of aggregate bandwidth.

5.1.2 Packet Drop Count

Followed by the throughput analysis we now analyze the effect of varying the buffer size on the cluster-head node. Figure 5.9 shows the number of packets plotted versus the varying the buffer size on the cluster-head node. The capacity of the buffer is varied according to the standard release of TinyOS [41] where the buffer size is varied between 2-16 bytes based on the application specifics. We compare our adaptive scheme's reliability versus a TDMA scheme that we simulate. In the TDMA scheme simulated for the scenario, the cluster-head nodes sets up a TDMA schedule and transmits the schedule to the nodes in the cluster. This ensures that there are no collisions and also allows the radio components of the non-cluster-head members to be turned off at all times except during their scheduled transmit times. This however causes greater energy wastage as nodes have to keep radios turned on even if they do not have any data to transmit during their schedule. Our adaptive scheme causes lower packets to be dropped for a given amount of simulation time (set to 1050 seconds). For any given buffer size our scheme drops a much lower number of packets as observed from the plot. The reduced number of packet drops in comparison to the TDMA based scheme occurs due to the control packets exchange during the self-configuration phase where the node ID's are exchanged and transmission of data-packets causes a reduction in the retransmissions.

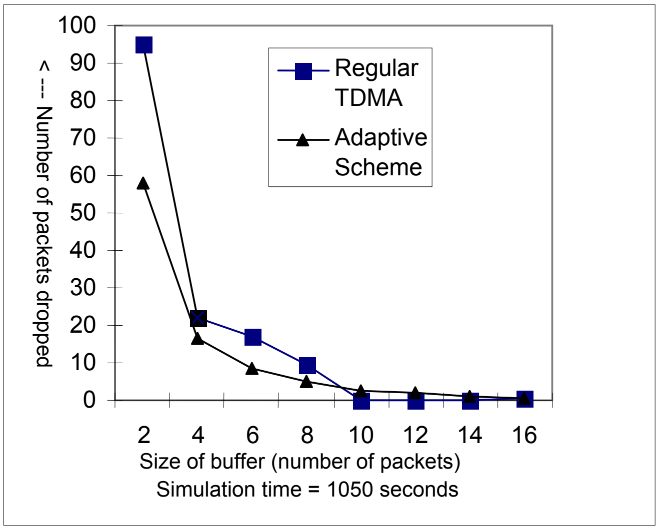


Figure 5.9: Number of packets dropped vs. Size of buffer

5.1.3 Energy Analysis

The energy curves in Figures 5.10 and 5.11, show plots of energy consumption with respect to varying traffic conditions, for any given node within a cluster. Traffic with an inter-arrival time of less than 3 seconds is considered heavy, while traffic with inter-arrival times greater than 7 seconds is considered light; and inter-arrival periods between 3 and 7 seconds correspond to medium traffic. From Figure 5.10, we see that though the energy consumption is somewhat high during higher traffic conditions, it reduces significantly with reduction in traffic. The low energy consumption for inter-arrival times greater than 3-4 seconds (medium- and light-traffic) is highly desirable in WSNs due to the nature of traffic experienced due to event-driven sensing where nodes in the sensor network are subject to extended times of minimal traffic over the network's lifetime. The higher energy consumption during heavy traffic can be attributed to the lesser sleeping

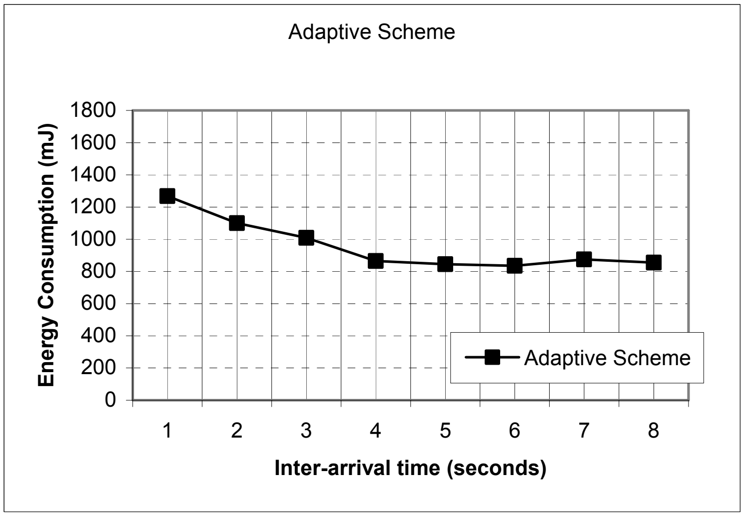


Figure 5.10: Energy consumption in a node within the cluster

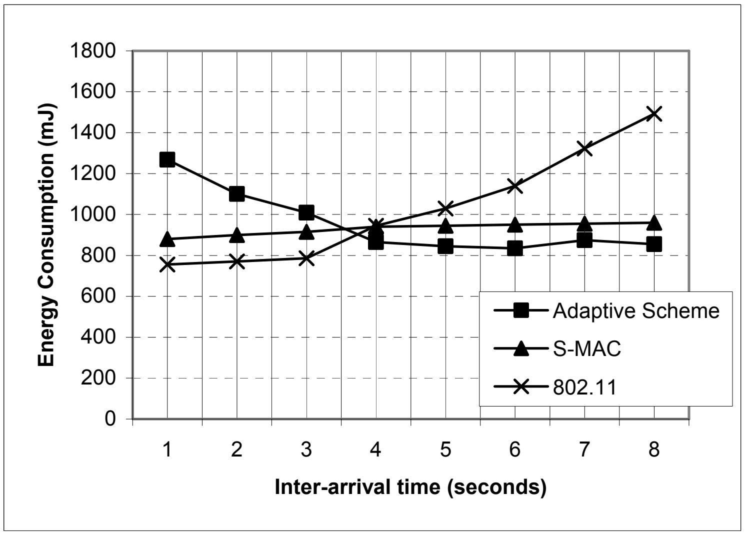


Figure 5.11: Energy consumption comparison charts

times of nodes within a cluster due to greater transmission times and increase in number of control packets. During medium- and light-traffic conditions, nodes spend greater time sleeping by setting their wakeup timer for longer times according to the adaptive learning pattern discussed in previous chapters.

The comparison curves in Figure 5.11 show that the S-MAC protocol and the IEEE 802.11 DCF have lower energy consumption for high traffic scenarios. However, the energy savings for higher traffic in the afore-mentioned protocols w.r.t our method is offset by the drastic reduction of energy during low and medium traffic conditions. While messages are fragmented as independent packets with RTS/CTS for each fragment in 802.11, more energy is saved by using shorter and lesser control packets in our method. As S-MAC uses message passing to transmit fragments in bursts, it burdens the nodes by the overhead of sending and receiving synchronization packets thereby introducing greater latency to transmit the same amount of data in comparison to our proposed method. Hence, for the case of medium- and light-traffic conditions our scheme has minimal energy consumption. Furthermore, as no specific node continues to remain the cluster-head over extended periods of time (following our proposed dynamic clustering approach), the energy consumption of any given node is found to be close to the energy consumed by any other node in its vicinity. Such a mechanism allows for a longer lifetime of the WSN.

6. Conclusions and Future Work

We conclude this thesis by summarizing the research discussed in the previous chapters, followed by a section on directions for future research. We have proposed a cross-layer design that exploits the characteristics of sensor networks to meet improved power-aware, scalable and fault-tolerant requirements of WSNs. To our knowledge, no current scheme attempts to satisfy all of these requirements because they all use general ad hoc network derived schemes centered around single-layer protocol designs.

6.1 Thesis Contributions

Our thesis presents a method for increased scalability and network wide energy-efficiency in WSNs. Each time the nodes in the network configure – new/mobile/hibernating nodes get discovered by the local search performed as a part of the dynamic clustering scheme – an interesting feature of our thesis – giving a relatively superior scalability capability to WSNs in comparison to existing procedures. By using the dynamic clustering approach, newly introduced/sleeping nodes that wake-up after extended periods of inactivity, can always be discovered. The scheme does not overload any specific node that becomes the cluster-head as cluster-heads move dynamically and hence change with time. By using the adaptive self-learning scheme for varying traffic more sleeping time ensures more network-wide energy savings. The savings in power is achieved by avoiding the overhearing effect through the elimination of the reception of all the packets inside the transmission range of the cluster, by putting nodes to sleep more often according to the self-learning traffic adaptive algorithm discussed in the previous chapters.

Based on the simulations that we have carried out to support our proposed methods, we observe that the throughput achieved is much higher for a given number of nodes within the cluster than throughput achieved by comparable schemes. The energy consumption using our scheme for low traffic is very low – this is very good as WSNs generally operate under event-driven detections, and traffic during the entire lifetime of the network is generally very low. Our method also reduces the overhead involved in the network for self-configuration by using smaller control packets. Thus, by allowing self-configuration using dynamic clustering and self-adaptability of the nodes within a cluster towards varying traffic, we show that it is possible to render more flexibility, scalability, longer lifetime and robustness in WSNs. To summarize, our cross-layer design for WSNs allows:

- nodes in the network to self-configure enabling ease of deployment,
- supports very high scalability of sensor nodes,
- higher throughput without much degradation of quality of sensed data, and
- improved energy-efficiency by adapting to varying traffic conditions inherent to WSNs.

6.2 Future Work

There is still much work to be done in the area of protocols for wireless sensor networks. The protocols developed in this research are for scenarios where the sensors have correlated data and nodes can adapt to the varying traffic of such scenarios only. However, there are important applications of WSNs where this is not the case. For example, sensor networks for medical monitoring applications may have different sensors located on and/or in the body to monitor vital signs. These networks will not be as large-scale as the ones we discussed, but they will have similar requirements to the sensor networks we discussed – long system lifetime, low-latency data transfers, and higher quality data. These networks will most likely focus on maximizing quality above all, and loss of information will not be acceptable. Therefore protocol architectures need to be developed that support QoS (Quality of Service) issues along with the unique considerations of these networks.

Obtaining a better understanding of application-perceived quality will enable new protocols to make intelligent trade-offs between energy and quality. This will be particularly useful when energy is highly-varying, such as in self-powered systems (e.g., systems that convert vibration into electrical energy). Protocols for self-powered systems will need to be adaptive to the current level of energy available and should produce the highest quality possible for a given energy to provide maximum benefit to the end-user. Such energy-quality scalability constraints add new parameters to the design of protocol architectures [10, 42]. Finally, it will be important to develop secure communication for WSNs. End-users need to be able to ensure unauthorized users cannot access the data from the sensor networks. Furthermore, end-users need to be able to authenticate the data. Application-specific solutions may be able to provide the level of security required without draining the node's limited energy. Without these security measures in place, the application of sensor networks will be limited.

The importance of mobility in sensor networks and schemes focused on the energy implications of node mobility (which none of the schemes have proposed to date) will also need to be addressed. Many applications for WSNs will require nodes to move as a group in a correlated fashion and follow typical mobility patterns. Schemes that address sensor network applications involving mobility will need to be considered. Possibly, a cluster based MAC conducive of handling group mobility patterns would be a good start-off point to research into node mobility. By making the self-learning process mobility aware it will be possible to avoid undesirable nodes during the formation of cluster thereby making the process more energy conserving.

As exemplified by our research in this thesis, cross-layer design will surely enable wireless networks of the future to support the services required by different applications, helping us get closer to the goal of “anytime, anywhere" communication among and between users and devices.

7. APPENDIX

APPENDIX I – Current Sensor Hardware

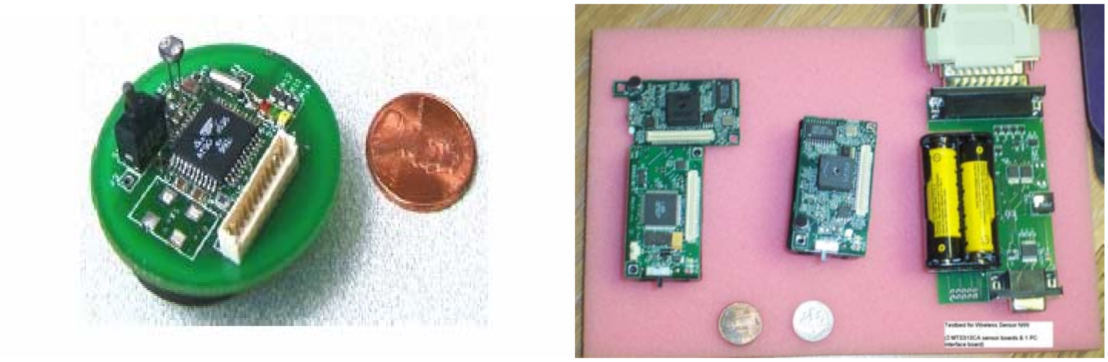


Figure 7.1: UC Berkeley motes (1st and 2nd generation)

UC Berkeley Smart Dust Motes: These nodes are designed by University of California, Berkeley and are popularly called smart dust. The mote is shown in Figure 7.1. This is a tiny node containing an MCU (ATMEL 90LS8535) with 8-bit Harvard architecture and 16-bit addresses. This controller provides 32 8-bit general purpose registers and runs at 4 MHz and 3.0 V. The system memory comprises 8KB flash memory and 512-byte SRAM as data memory. The radio is an asynchronous input/output device with hard real-time constraints. It consists of an RF Monolithics 916:50 MHz transceiver, antenna and collection of physical-layer components to configure the physical layer characteristics such as signal strength and sensitivity. It comes with a temperature sensor with an option to mount custom-selected sensors on the sensor board. The nodes run TinyOS operating system which fits in 178 bytes of memory supporting two-level scheduling and allows for high concurrency to be handled in a very small amount of space.

(a) PC-104 based sensor node

(b) WINS NG 2.0 sensor node

Figure 7.2: PC-104 and Sensoria nodes

PC-104 based Nodes: PC-104 is an industry standard of PC-compatible modules that can be stacked together to form a custom-designed embedded system. The term PC-104 is derived from the connector used to stack different boards having 104 pins. The standard was initially released in 1992. Since these systems are made with hardware compatible with PC systems it is easy to configure them along with the PCs. The PC-104 sensor nodes are custom built with chosen processor, memory configuration and hard disk. The SCADDS testbed of USC/ISI consists of 30 nodes built using PC-104 based products. Figure 7.2 (a) shows a PC-104 based node.

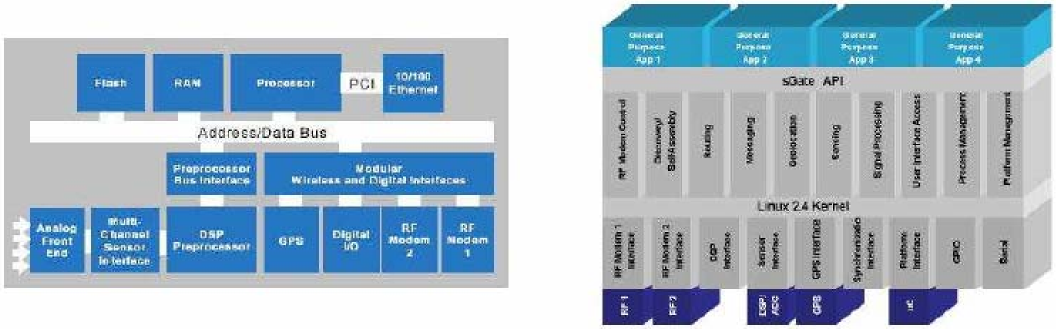


Figure 7.3: Hardware and software architectures of WINS NG 2.0 node

Sensoria WINS NG nodes: WINS NG [23] node is a Linux based embedded computing platform with several interfaces to externally connect sensors, wireless extension cards and any serial port devices. Figure 7.2 (b) shows a Sensoria sensor node. This node uses the Hitachi SH-4 processor running at 167 MHz. The SH-4 is a 32-bit RISC with a 128-bit vector floating point unit (FPU) and super-scalar implementation providing higher speeds at low clock rates. The sensor node supports four sensors and also hosts a GPS module for geo-location information of the nodes. The node communicates with the dual RF modems built-in, both of them in the 2:401 - 2:495 GHz ISM band using frequency-hopping spread spectrum (FHSS). The hardware architecture of the node is seen in the left side of Figure 7.3 and the software architecture of the node in right side of Figure 7.3. These nodes run Linux kernel 2:4:16 and all the hardware used in the node is supported by the kernel. The SH-4 cross-compilers are used to compile the code written onto these nodes. Sensoria provides the API required for RF modem control and data acquisition.

APPENDIX II – AVERAGE INTER-ARRIVAL TIME OF THE SYSTEM

An exponentially distributed random variable X ~ exp (λ) has the following probability density function.

$$f_x(x) = \lambda e^{-\lambda x} \quad \mathrm{x} > 0 \tag{a4\_1}$$

The mean of X can be calculated as;

$$\mathrm{E}\{\mathrm{x}\} = \int_0^\infty \mathrm{x} f_x(x)dx = \frac{1}{\lambda} \tag{a4\_2}$$

The probability that X is greater than x + y given that X is greater than α is:

$$P\{X > \alpha\} = \int_\alpha^\infty \mathrm{x} f_x(x)dx = \mathrm{e}^{-\lambda\alpha} \tag{a4\_3}$$

Now, assume that we have N independent nodes sending traffic with exponentially distributed inter-arrival times, which are represented by random variables X_i i = 1... N. The first node to be handled in the traffic model of the simulator is the one which has the minimum of those inter-arrival times, which is also a random variable. Therefore, we can express the system inter-arrival time of N users as;

$$X_{\min} = \min\{X_1, X_2, \ \dots \ , X_N\} \tag{a4\_4}$$

The probability that $X_{\min}$ being greater than x is defined as;

$$P\{X_{\min} > x\} = P\{X_1 > x, X_2 > x, \ \dots \ , X_N > x\} \tag{a4\_5}$$

Since, X_i's are assumed to be independent, and then the above expression can be written as;

$$P\{X_{\min} > x\} = P\{X_1 > x\} P\{X_2 > x\} \dots P\{X_N > x\} \tag{a4\_6}$$

Using (a4_3), we can express (a4_6) as;

$$P\{X_{\min} > x\} = \mathrm{e}^{-\lambda \mathrm{x}} \mathrm{e}^{-\lambda \mathrm{x}} \dots \mathrm{e}^{-\lambda \mathrm{x}} = \mathrm{e}^{-\lambda N \mathrm{x}} \tag{a4\_7}$$

Comparing (a4_3) with (a4_7), we see that X_{min} is also exponentially distributed random variable with λN as parameter. To determine the mean inter-arrival time for a cluster with N non-cluster-head members with identical exponential distributions, we refer to (a4_2) where λN as a parameter. We see that mean inter-arrival time for the cluster is N times less than that of a single node.

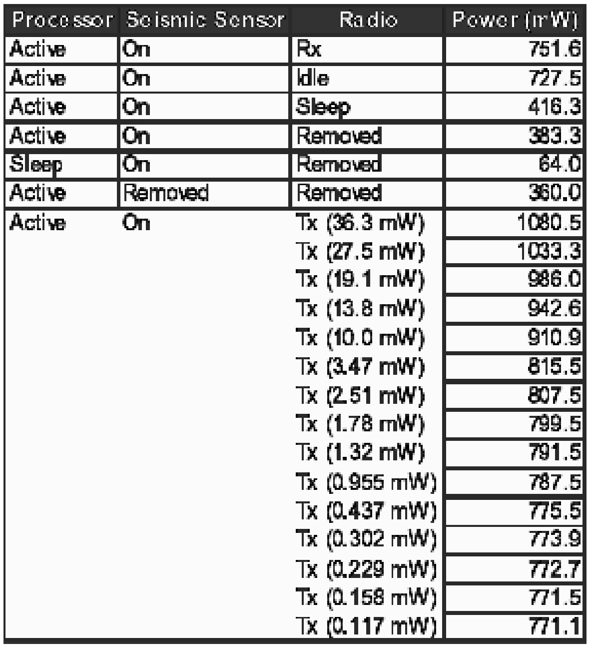
APPENDIX III – THE HEAVY TAILED DISTRIBUTIONS

Let X be a random variable with cdf F(x) = P[X $\le$ x] and complementary cdf (ccdf)

$$\bar{F}(x) = cx^{-\alpha} \quad 0 < \alpha < 2 \text{ as } x \to \infty$$

If F(x) is heavy tailed, then X shows very high variability. In particular X has infinite variance for $0 < \alpha < 2$ and infinite mean if $0 < \alpha \le 1$. If $\{X_i$, i = 1, 2, ...$\}$ is a sequence of observations of X then the sample variance of $\{X_i\}$ as a function of i will tend to grow without limit, as will the sample mean if $\alpha \le 1$.

APPENDIX IV – Data sheets



Processor	Seismic Sensor	Radio	Power (mW)
Active	On	Rx	751.6
Active	On	Idle	727.5
Active	On	Sleep	416.3
Active	On	Removed	383.3
Sleep	On	Removed	64.0
Active	Removed	Removed	360.0
Active	On	Tx (36.3 mW)	1080.5
		Tx (27.5 mW)	1033.3
		Tx (19.1 mW)	986.0
		Tx (13.8 mW)	942.6
		Tx (10.0 mW)	910.9
		Tx (3.47 mW)	815.5
		Tx (2.51 mW)	807.5
		Tx (1.78 mW)	799.5
		Tx (1.32 mW)	791.5
		Tx (0.955 mW)	787.5
		Tx (0.437 mW)	775.5
		Tx (0.302 mW)	773.9
		Tx (0.229 mW)	772.7
		Tx (0.158 mW)	771.5
		Tx (0.117 mW)	771.1

Processor/Radio Board MPR300CA Specifications

Speed 4MHz
Flash 128K bytes
SRAM 4K bytes
EEPROM 4K bytes
Serial Comms UART
A/D 10 bit ADC 8 channel
Processor Current Draw 5.5 mA active current, typ
<1uA sleep mode, typ
Serial Flash 4Mbit permanent ID 64 bits
Radio Frequency 916 MHz ISM band
Data Rate 50 Kbits/sec max
Power 0.75 mW
Radio Current Draw 12mA transmit current, typ
1.8 mA receive current, typ
5uA sleep current, typ
Radio Range 200 feet programmable
Power 2X AA batteries attached pack
External Power 3 Volts connector provided
User Interface 3 LEDs user programmable
Expansion Connector 51 pin connector for plug-in sensor boards

References specific to data sheets:

1. www.xbow.com/Support/Support_pdf_files/ MTS-MDA_Series_User_Manual_RevB.pdf

2. Stemm, M. and Katz, R. “Measuring and reducing energy consumption of network interfaces in hand-held devices”. Institute of Electronics, Information, and Communication Engineers Transactions on Communications, E80B (8):1125--1131, August 1997.

References

[1] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for smart dust, Proceedings of the ACM MobiCom'99, Washington, USA, 1999, pp. 271–278.

[2] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A.Chandrakasan, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp. 272–286.

[3] S. Singh and C.S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," ACM Computer Communication Review, vol. 28, no. 3, pp. 5–26, July 1998.

[4] *V. Bharghavan, A. Demers, S. Shenker, and L. Zhang.* MACAW: A Media Access Protocol for Wireless LANs, In Proceedings of SIGCOMM '94 (University College London, London, U.K., September 1994), pp. 212-225.

[5] http://robotics.eecs.berkeley.edu/~pister/SmartDust

[6] *V. Raghunathan, C. Schurgers, Park.S, and M.B. Srivastava*, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, Volume: 19 Issue: 2 , March 2002 Page(s): 40 –50.

[7] *Vardhan, S.; Wilczynski, M.; Portie, G.J.; Kaiser, W.J.*; **"**Wireless integrated network sensors (WINS): distributed in situ sensing for mission and flight systems" Aerospace Conference Proceedings, 2000 IEEE, Volume: 7 , 18-25 March 2000, 459 -463 vol.7.

[8] http://www.gigascale.org/picoradio/pdf/IIB_TechRation.pdf

[9] Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in theEnergy-Latency-Density Design Space", IEEE Transactions on Mobile Computing, vol.1, no.1, January-March 2002.

[10] R. Powers, "Advances and Trends in Primary and Small Secondaey Batteries," IEEE Aero. Elec. Sys. Mag., vol. 9, no. 4, Apr. 194 pp. 32-6.

[11] LAN-MAN Standards Committee of the IEEE Computer Society, "Wireless LAN medium access control(MAC) and physical layer(PHY) specification", IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.

[12] J. C. Haartsen, *The Bluetooth radio system*, IEEE Personal Communications Magazine, p.28-36, February 2000.

[13] D. Amis et al., *Max-min D-Cluster Formation in Wireless Ad Hoc networks*, IEEE INFOCOM, March 2000.

[14] M. Gerla and J. T. Tsai, *Multicluster, Mobile, Multimedia Radio Network*, Wireless Networks, 1995, pp. 255-65.

[15] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for self-organization of a wireless sensor Network", IEEE Personal Communications, October 2000, pp. 16–27.

[16] S. S. Raghuwanshi, A. Mishra, "An Approach Towards Improved Energy-efficiency in Wireless Sensor Nets", IEEE RAWCON 2003.

[17] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks", in *Proceedings of the Hawaii International* Conference on Systems Sciences, Jan. 2000.

[18] *Okino, C.M.; Corr, M.G.;* "Best effort adaptive routing in statistically accurate sensor networks" Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on , Volume:1, 12-17 May 2002, Page(s): 345 -350.

[19] *Elson, J.; Estrin, D.;"*Time synchronization for wireless sensor networks", Parallel and Distributed Processing Symposium., Proceedings 15th International , 23-27 April 2001.

[20] *Sichitiu, M.L.; Veerarittiphan, C.;"*Simple, accurate time synchronization for wireless sensor networks", Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE , Volume: 2 , 16-20 March 2003.

[21] C. Guo, L. C. Zhong, and J. M. Rabaey, "*Low-Power Distributed MAC for Ad Hoc Sensor Radio Networks*, Proc. Internet Performance Symp. (Globecom 2001), November 2001.

[22] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci Wireless sensor networks: a survey Computer Networks, 38 (4) (2002) pp. 393-422.

[23] J.G. Pottie; W.J. Kaiser. Wireless integrated network sensors. Comm. ACM, 3(5):51--58, May 2000.

[24] Zhong, L.C.; Jan Rabaey; Chunlong Guo; Rahul Shah, Data link layer design for Wireless sensor networks Military Communications Conference, 2001. MILCOM2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE, Volume: 1, 2001.

[25] Y. Tseng, S. Wu, C. Lin, J. Sheu, A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks, IEEE International Conference on Distributed Computing Systems, Mesa, AZ, April 2001, pp. 419–424.

[26] S. Wu, C. Lin, Y. Tseng, J. Sheu, A new multi channel MAC protocol with on-demand channel assignment for multihop mobile ad hoc networks, International Symposium on Parallel Architectures, Algorithms, and Networks, I-SPAN 2000, Dallas, 2000, pp. 232–237.

[27] Wei Ye, John Heidemann, Deborah Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, IEEE Infocom 2001.

[28] W.B.Heinzelman, "Application-specific Protocol Architectures for Wireless Networks," *Thesis*, Ph.D., MIT June 2000.

[29] T. Rappaport, Wireless Communications: Principles and Practice, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[30] Woo, D. Culler, A transmission control scheme for media access in sensor networks, Proceedings of ACM MobiCom'01, Rome, Italy, July 2001.

[31] Wireless Today: Forecast of Users with Wireline Versus Wireless High-Speed (Individual) Access, North America, 1998-2003. http://www.wirelesstoday.com/, January 06, 1999.

[32] L. Robinson. Japan's new mobile broadcast company: Multimedia for cars, trains, and hand helds. In Advanced Imaging, pages 18-22, July 1998.

[33] J. Agre and L. Clare. An Integrated Architecture for Cooperative Sensing Networks. Computer, 33(5):106-108, May 2000.

[34] Chandrakasan, R. Amirtharajah, S.-H. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang. Design Considerations for Distributed Microsensor Systems. In IEEE 1999 Custom Integrated Circuits Conference (CICC), pages 279-286, May 1999.

[35] MIT's Magazine of Innovation, Technology Review, 10 Emerging Technologies that will change the world: Wireless Sensor Networks, February 2003, pp. 33-7.

[36] http://www.circlemud.org/~jelson/writings/timesync/

[37] K. Pahlavan and A. Levesque. Wireless Information Networks. John Wiley & Sons, Inc., New York, 1995.

[38] S. S. Raghuwanshi, A. Mishra, "A self-adaptive clustering based algorithm for increased Energy-efficiency and Scalability in Wireless Sensor Networks", Proceedings IEEE Vehicular Technology Conference 2003, VTC 2003, October 6-9, Florida.

[39] UCB/LBNL/VINT Network Simulator – ns (Ver 2+), http://www.isi.edu/nsnam/ns/

[40] Oliver Kasten, *Energy Consumption*, Eldgenossische Technische Hochschule, Zurich. http://www.inf.ethz.ch/kasten/research/bathtub/energy_consumption.html

[41] http://webs.cs.berkeley.edu/tos/

[42] http://www.instat.com/abstracts/md/2000/md0003md_abs.htm

[43] http://www.cisco.com/warp/public/779/govtaffs/factsNStats/broadbanddeployment.html

[44] http://users.ece.gatech.edu/~grimace/research/sensorsimii/

[45] C. Perkins, Ad Hoc Networks, Addison-Wesley, Reading, MA, 2000.

[46] P. Karn, "*MACA – A new channel access method for packet radio*," in RRL/CRRL Amateur Radio 9th Computer Networking Conference, pp.134-140, 1990.

[47] IEEE, Wireless LAN medium access control (MAC) and physical layer (PHY)Spec." IEEE 802.11 standard, 1998.

[48] A. Colvin, "CSMA with collision avoidance," *Computer Communications*, vol. 6, no. 5, pp. 227–235, 1983.

[49] H. Woesner, J.-P. Ebert, M. Schlager, and A. Wolisz, "Power-saving mechanisms in emerging standards for wireless LANs: the MAC level perspective," *IEEE Personal Communications*, vol. 5, pp. 40–48, June 1998.

VITA

Srajan Singh Raghuwanshi was born in Bhopal, India in 1980. His interests cover a diverse area in Electrical and Computer Engineering ranging from Networking, Wireless Communications, to System design and Administration. He graduated from Osmania University, India in 2001 with a Bachelor of Engineering degree in Electronics and Communications Engineering. As an undergraduate engineering student, he (as a part of a team of three) designed and implemented the design of a robotic vehicle similar to the MARS-ROVER that has been used for Mars space exploration.

Mr. Raghuwanshi joined Virginia Tech in Fall 2001, as a graduate student, in the Electrical Engineering Department. He has been working on energy-efficiency at the data-link and network layer since Fall 2002 and has proposed a novel cross-layer design as a part of his thesis work. The graduate committee has decided and recommended that he should file for a disclosure and patent this fine piece of work. Along with being a good researcher, he has also been helping the Virginia Tech Research Division as a Systems Administrator for server support and networking related issues. A few current publications of his include,

1. "**An Approach Towards Improved Energy-efficiency in Wireless Sensor Nets**", IEEE Radio and Wireless Conference 2003, RAWCON 2003, August 10-13, Boston MA.
2. **"A self-adaptive clustering based algorithm for increased Energy-efficiency and Scalability in Wireless Sensor Networks**", Proceedings IEEE Vehicular Technology Conference 2003, VTC 2003, October 6-9, Florida.

Mr. Srajan successfully defended his Master's thesis on August 29th, 2003 and currently, (as of October 2003) he is associated with AutoGenomics ©, a biotech firm in San Diego, California, USA as a Systems Engineer. He can be contacted at shanoosrajan@yahoo.com