A CDMA-Based, Self-Organizing, Location-Aware Media Access Control Protocol

Bao Hua Liu, Nirupama Bulusu*, Huan Pham, Sanjay Jha

Network Research Laboratory
School of Computer Science and Engineering
The University of New South Wales

National ICT Australia Ltd.*

Email: {mliu, nbulusu, huanp, sjha}@cse.unsw.edu.au

UNSW-CSE-TR-0401

January 11, 2004

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY · AUSTRALIA

Abstract

In this paper, we propose CSMAC, a novel CDMA-based, self-organizing, location-aware media access control (MAC) protocol for sensor networks. We argue that no single MAC protocol is suitable for all sensor network applications, which cover a broad range of application domains from wildlife tracking to real-time battlefield surveillance. Previously proposed MAC protocols for sensor networks such as S-MAC [12] primarily prioritize energy-efficiency over latency. Our protocol design balances the considerations of energy-efficiency, latency, accuracy, and fault-tolerance in sensor networks. CSMAC uses Code Division Multiple Access to reduce channel interference and consequently message latency in the network. It exploits location awareness to improve energy-efficiency by employing two special algorithms in the network formation process — Turn Off Redundant Node (TORN) and Select Minimum Neighbor (SMN). ns-2 simulations show that in a 10-hop network topology, CSMAC can achieve upto 74% lower mean latency than SMAC, while consuming 41% lower mean energy per node.

1 Introduction

Continuing advances in Micro-Electro Mechanical Systems (MEMS) enable the construction of a wide variety of sensors/actuator devices. These sensors/actuators consist of one or more sensing units, embedded microprocessors and low power radios. Sensors are normally untethered and powered by batteries. They communicate over short distances using wireless media. Therefore, a large number of distributed sensors can autonomously organize themselves into a multi-hop wireless sensor network. Due to the broad range of potential applications of such sensor networks, they have garnered much academic and industrial attention in recent years.

1.1 Motivation

Because the design of an effective media access control (MAC) protocol is one of the fundamental communication challenges in sensor networks, it has been previously addressed in [12, 3, 5, 4]. However, sensor network applications span a broad range of domains from wildlife tracking to real-time battlefield surveillance. In the wildlife context where sensor data is being collected for scientific research, the network may be inherently delay-tolerant. Whereas in a battlefield application where sensor data may be used to detect land mines, alert soldiers of the detection of enemy convoy vehicles etc., accurate and timely delivery of sensed data may mean the difference between life and death. Therefore, we believe that no single MAC protocol is suitable for all sensor network applications.

Previously proposed MAC protocols for sensor networks such as S-MAC [12] primarily prioritize energy-efficiency over latency. While it is true that in some cases latency is not a critical factor for some applications (e.g., data collection for scientific research), many applications may have stringent latency requirements (e.g., battlefield, real-time monitoring of bush fires). In the battlefield, a delay of one second sometimes means life or death. It is also unacceptable that the sensor network delayed an imminent bushfire alert because several intermediate sensors are sleeping or waiting their turn to transmit. Latency is an application dependent criteria that cannot be ignored. Moreover, previous MAC protocols have not considered incorporating sensing accuracy requirements of the application to influence the formation of the network topology.

Our goals for the design of the MAC protocol are the following:

- *Energy Efficiency*: As sensor nodes are normally battery-powered, the MAC protocol must be energy efficient so as to maximize not only the lifetime of the individual node but also the lifetime of entire system.
- *Low Latency*: The observer is interested in knowing about the phenomena within a given delay.
- *Sensing Accuracy*: Obtaining accurate information is the primary objective of the observer, where sensing accuracy is an application dependent factor. The network efficiency can be further improved if network organization is based on sensing needs.
- *Fault Tolerance*: The sensor network must be fault-tolerant so that non-catastrophic failures are hidden from the application.
- *Scalability*: Sensor network applications often feature a large number of sensor nodes. Therefore, the MAC protocol must be scalable.

1.2 Design Rationale

It is challenging to design a media access control protocol suitable for low-latency and accurate delivery of sensed information that is also fault-tolerant, scalable, and energy-efficient. We comment next on the key features of our MAC protocol that enable it to achieve these goals:

1.2.1 Self-organizing and Adaptive

To be scalable and fault-tolerant, our media access control protocol is self-organizing and adaptive. Rather than relying on manual configuration, network formation and maintenance is based on nodes dynamically discovering and selecting which neighbors to communicate with in order to form a connected multi-hop network topology. Because nodes directly communicate with only a limited subset of their neighbors, it also scales well.

1.2.2 Location-Aware

To be energy-efficient, our MAC protocol exploits the location-awareness of sensor nodes during network formation in two algorithms — *Turning off Redundant Node (TORN)* and *Select Minimum Neighbor (SMN)*.

Sensing accuracy is an application dependent criteria. Our *Turning Off Redundant Node (TORN)* algorithm tries to improve energy-efficiency while maintaining the application-desired sensing accuracy. There are some application scenarios where re-deployment of new nodes in a current sensor network is unlikely or impossible. Therefore, sensor nodes will be normally deployed with high redundancy. Some researchers have proposed solutions to efficiently exploit redundancy at the network layer through in-network data processing. Our approach is to turn off all redundant nodes instead of using them. Redundant nodes not only generate unnecessary sensing data, but also need to transmit and process these data thereby wasting limited battery power. Redundant nodes also cause interference to their neighbors. We believe keeping redundant nodes active will have negative impact on sensor network performance.

We propose a new concept called *Sensing Resolution (SR)*. SR denotes the sensing precision and is an application level criteria. For example, assume two meter is enough for sensing resolution for a given application. If two nodes are accidentally deployed within a distance less than two meters, keeping them both active is unnecessary. In this case, our TORN algorithm forces one node to turn itself off completely. The inactive node provides fault tolerance to the active node when it fails. In this way, we can achieve an evenly deployed sensor network even when they are deployed unevenly. By completely turning off (with *completely*, we mean both the radio and sensing unit are turned off, only a very low power clock is running to wake up the node sometime in the future) redundant nodes, the battery power of these redundant nodes are preserved for future use, prolonging the network lifetime as a whole.

To prolong the lifetime of an individual node, we use *Select Minimum Neighbor (SMN)* algorithm to minimize the number of neighbors of a node and prohibit a node communicating directly with its further away neighbors, even if they are within the radio range. As sensors are normally deployed in harsh outdoor environment, the path loss exponent for sensor ground-lying antenna, in 800-1000MHz band, is usually close to four [6]. This prompted us to ask a fundamental question in sensor network: *Who is a node's neighbor*? Figure 1 shows three nodes that are randomly deployed in the sensing field. Is node C a neighbor of node A? Remember radio propagation loss is directly proportional to the fourth exponent of distance. If the energy required for direct transmission from node A to node C is larger than the cumulative energy required for the multi-hop path from A to B to C, then node C is not selected as a neighbor of node A. Our SMN algorithm is based on this propagation loss analysis.

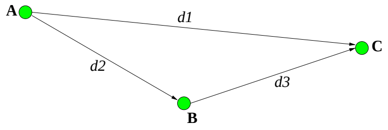

Figure 1: *C is not selected as a neighbor of A*

1.2.3 CDMA-Based

Because we use multi-hop instead of direct long-range communication for energy efficiency, we increase network latency. To compensate for the latency sacrifice, we use CDMA spread spectrum approach to improve latency performance. With CDMA, all nodes can transmit in the same frequency band simultaneously. Channels are distinguished by pseudo noise code (PN code) rather than the time slot in TDMA and frequency band in FDMA.

Simple FDMA approach can not be used because it is very difficult for a node to decide when to synchronize to which frequency. With TDMA, each node must wait for its turn (time slot) to transmit. TDMA also requires stringent time synchronization between a node and its neighbors. Comparatively, CDMA requires less stringent timing issues versus TDMA. CSMA is suitable for low traffic (less collision) scenarios as the media is shared by all sensor nodes. Because sensors are most likely to transmit data simultaneously when an event occurs, CSMA is not a good choice with its backoff transmission approach. Collided packets must be discarded and energy is wasted. CSMA also suffers from *hidden terminal* problem. Use of RTS, CTS and ACK control packets in CSMA causes significant overheads for short sensor data packets.

By employing CDMA technique, our MAC protocol design is contentionless and the communications are peer to peer. Each node can transmit to its neighbors without using any control packet exchange such as IEEE 802.11 RTS, CTS, ACK, etc. Although these control packets are small, the data packets in sensor networks are also small. Elimination of these control packets exchange will reduce the radio energy consumption significantly.

Because CDMA increases the energy expended by the coding and decoding electronics, our protocol must be based on low power CDMA modem technology. Chien et al [8] designed a low-power DSSS (Directed Sequence Spread Spectrum) modem architecture on an FPGA prototype that shows a power dissipation of 33mW. When implemented using CMOS ASIC technology, the estimated power is less than 1mW. We believe with the advances in technology, the electronic power consumption will continue to decrease, and the ratio between the electronic power consumption and the radio amplifier power consumption will also decrease because the radio signal power is more distance dependent and less technology dependent.

Organization — The rest of this paper is organized as follows. Section 2 describes related work. Section 3 presents the technical background necessary for understanding our protocol design. Section 4 elaborates the protocol design and network formation process. Section 5 provides simulations and analysis. Section 6 discusses several further improvements and research directions. We conclude our paper in Section 7.

2 Related Work

Media access control protocols for ad hoc sensor networks have been previously addressed [12, 3, 5, 4]. These protocols can be broadly classified into *contentionless* and *contention-oriented*. Contentionless MAC protocols for sensor networks are based on Frequency Division Multiple Access (FDMA) or Time Division Multiple Access (TDMA) approaches. Contention-oriented MAC protocols are adapted from the IEEE 802.11 standard.

SMACS (Self-Organizing Media Access Control for Sensor Networks) [5] is a distributed protocol which enables a collection of nodes to discover their neighbors and establish transmission/reception schedules for communicating with them without the need for any local or global master nodes. Each node maintains a TDMA-like frame called superframe, in which the node schedules different time slots to communicate with its known neighbors. In SMACS, neighbor discovery and channel assignment phases are combined so that by the time nodes hear from all their neighbors, they would have formed a connected network. Although network wide time synchronization is not necessary, the communicating neighbors in a subnet still need to be time-synchronized. Power conservation is achieved by turning off the radio during idle time slots. Unlike our protocol, network formation in SMACS is not location-aware. A node may select a neighbor that is further away from it instead of a near by neighbor. Because radio propagation loss is directly proportional to the fourth exponent of distance, it is not energy-efficient. Moreover, if the TDMA time slots of two neighbor nodes do not overlap, these two nodes will disengage from each other. By using a TDMA approach, a node must wait its turn to transmit to a specific neighbor even if the channel is idle. And this waiting time can accumulate along the multi-hop route from source to sink.

Woo and Culler [4] proposed a CSMA-based MAC protocol, designed especially to support the periodic and highly correlated traffic of some sensor network applications. They propose an adaptive transmission rate control (ARC) scheme, whose main goal is to achieve media access fairness by balancing the rates of originating and route-through traffic. Because it is CSMA-based, this approach may suffer from control overheads and hidden terminal problems.

Shih *et al*[3] investigate the impact of non-ideal physical layer electronics on MAC protocol design for sensor networks and proposed a centrally controlled MAC scheme. While a pure TDMA scheme dedicates the full bandwidth to a single sensor node, a pure FDMA scheme allocates minimum signal bandwidth per node. Pure TDMA is not always preferred due to the associated time synchronization cost. The hybrid TDMA/FDMA scheme they propose optimizes the power consumption of the transceiver and results in lowering the overall power consumption of the system.

Inspired by PAMAS [9], SMAC (Sensor-MAC) [12] is based on IEEE 802.11 standard but improves the energy inefficiency of 802.11. SMAC identified several major sources of energy waste including *collision, overhearing, control packet overhead, and idle listening.* SMAC uses IEEE 802.11 CSMA/CA approach to avoid collisions. To avoid overhearing, SMAC puts a node to sleep when a neighbor node is transmitting. A scheduled periodic sleep and listening pattern is used to decrease the idle listening energy consumption. The main drawback of SMAC is high message delivery latency as SMAC is designed to sacrifice latency for energy savings. Actually, the assumption of SMAC design is that applications have long idle periods and can tolerate latency in the order of network messaging time.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [10] is a cross-layer protocol design, encompassing MAC and network layers. It provides a combined TDMA/CDMA based MAC approach. Each node communicates with a dynamically elected cluster head directly (no multi-hop) using TDMA scheme. Cluster heads communicate with remote destination (sink) directly using a CDMA approach. In contrast, our MAC protocol relies exclusively on CDMA techniques to reduce overall network interference.

3 Background

This section presents the technical background for our protocol design, focusing on radio propagation and code division multiple access (CDMA).

3.1 Radio Propagation

Neighbor selection for energy-efficient network formation in our MAC protocol is motivated by radio propagation behavior. The mechanisms behind electromagnetic wave propagation are diverse, but can generally be attributed to reflection, diffraction, and scattering [19]. The free space propagation model is given by Friss equation as shown below:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{1}$$

where P_t is the transmitted power, $P_r(d)$ is the received power which is a function of the transmitter-receiver (T-R) separation, G_t is the transmitter antenna gain, G_r is the receiver antenna gain, d is the T-R separation distance in meters, L is the system loss factor ($L \geq 1$), and λ is the wavelength in meters. The above equation reveals that the radio propagation loss in free space is directly proportional to the square of distance. A single line-of-sight path between two sensors is seldom the only physical means for propagation. The two-ray ground reflection model considers both the direct path and a ground reflect propagation path between transmitter and receiver. The received power at distance d is predicted by

$$P_r(d) = \frac{P_t G_t G_r (h_t)^2 (h_r)^2}{d^4 L} \tag{2}$$

where h_t and h_r are the heights of transmit and receive antenna respectively. Above equation shows a faster power loss than free space equation 1. Sohrabi et al [6] have shown that for ground-lying antennas (which is normally the case for sensor nodes), in 800-1000 MHz frequency band, the average range of path loss exponent is between 2.2 to 5. This motivates that multi-hop communication should be used instead of direct communication between sensor nodes for energy efficiency.

The goal of our *Select Minimum Neighbor (SMN)* algorithm is to minimize the number of neighbors of a node and only allow a node to communicate directly with its limited neighbors only when there is no alternative multi-hop path that can provide a less energy consuming route. The *SMN* algorithm will be discussed in Section 4.3.

3.2 Code Division Multiple Access

We use CDMA spread spectrum approach for the following reasons:

- With CDMA, all nodes can transmit in the same frequency bandwidth simultaneously.
- CDMA requires less stringent time synchronization compared to TDMA.[1].
- CDMA is *contentionless* and therefore does not need RTS/CTS/ACK control packet exchange between sensor nodes.
- Spreading spectrum implies the reduction of multi-path effects and good anti-jam performance. Data spreading results in greater immunity to radio frequency interference as compared to narrow-band signaling. And that is why IEEE 802.11b standard also employs DSSS (Direct Sequence Spread Spectrum) for better channel performance.

---

[1] CDMA does need synchronization of the locally generated code signal and received signal. This code synchronization is accomplished at the receiver side.

- CDMA requires stringent power control, which is a positive feature for power saving purpose for sensors. The inherent static characteristic of sensor node makes the power control less complex than CDMA cellular system.
- The generation of PN code is simple, several shift-registers are enough. The coded signal is simply multiplication of base band signal and PN code with DS-CDMA.
- Because receiver use the same frequency to receive, the frequency synthesizer is simple.
- Coherent demodulation of spread spectrum signal is possible.

Although CDMA system increases the coding and decoding energy consumption, the spreading of baseband signals gives a much better channel relative to non-spreading signals. Spread spectrum (SS) system can use less transmission power to achieve the same bit-error-rate (BER) performance when compared to a non SS system.

3.2.1 Pseudo Noise Codes

CDMA uses pseudo-random noise code (PN code or PN sequence) to differentiate channels. A PN code is a sequence of chips valued -1 and 1 (polar) or 0 and 1 (non-polar). Such bit-sequences have noise-like properties like spectral flatness and low cross and auto correlation values, and thus complicate jamming or detection by non-target receivers [20]. In practice, the PN sequence selected must have both good cross-correlation and good auto-correlation because the receiver uses cross-correlation to separate the appropriate signal from other interfering signals (those destined for other receivers), and uses auto-correlation to reject multi-path interference. Specific PN codes include Walsh codes, M-sequences, Gold codes and Kasami codes. Codes can be orthogonal or non-orthogonal. Walsh code sets are orthogonal. They are generated by the *Hadamard matrix* expansion as shown below:

$$w_{2n} = \begin{vmatrix} w_n & w_n \\ w_n & \bar{w_n} \end{vmatrix} \tag{3}$$

with the seed matrix

$$w_1 = 0 \tag{4}$$

3.2.2 Processing Gain

The main parameter of a CDMA spread spectrum system is the processing gain, which is defined as the ratio of spread data rate to the initial data rate:

$$G_p = \frac{B_t}{B_i} \tag{5}$$

where B_t is the transmission bandwidth, and B_i is the bandwidth of the information (baseband) signal. For example, IS-95 defines the original base band speech data rate of 9600 bps, after convolution coding (for FEC, forward error correction) and Walsh coding (spreading), the data rate is *spreaded* to 1.2288 Mbps. So the processing gain is $G_p = 1.2288Mbps/9.6kbps = 128 = 21dB$. Processing gain determines the number of simultaneous transmissions that can be allowed in a system. CDMA signal experiences high levels of interference as all other CDMA coded signals are treated as noise. The total interference also includes background white noise and other spurious signals. When the signal is received, the correlators at the receiver use the processing gain to extract the desired signal out of the noise. For CDMA spread spectrum systems it is advantageous to have a processing gain that is as high as possible to allow many simultaneous transmissions. Fortunately, we are not expecting a sensor node to have as many neighbors as a CDMA cellular system.

Our *Select Minimum Neighbor* algorithm even minimizes the number of neighbors. A node only communicates directly with its neighbors only when *there is no other neighbor node that can provide a multi-hop path with less energy consumption*. This means that we may achieve similar channel characteristics (e.g., BER) with possibly lower processing gain. We will discuss *Select Minimum Neighbor* algorithm in Section 4.3.

3.2.3 Spread Spectrum Techniques

Different spread spectrum techniques exist: Direct-Sequence (DS), Frequency-Hopping (FH), Time-Hopping (TH) and Multi-Carrier CDMA (MC-CDMA). It is also possible to combine these techniques.

DS-CDMA is the most popular technique and easy to implement because the baseband signal is simply multiplied with a PN sequence. The main problem with DS-CDMA is the so-called *near-far* problem. Suppose two nodes transmit with the same power, and the interfering transmitter is relatively much closer to the receiving node than the intended transmitter. In this case, the correlation in the receiver of the signal received from the interfering transmitter may exceed that of the signal received from the intended transmitter (*i.e.*, the correct code). As a result the intended data is hard to recover. The solution to overcoming the near-far problem is to use stringent power control so as to ensure that the received signals always have the same amount of power at the receiving node. Fortunately, sensor nodes are normally static instead of mobile, and the power control is much easier compared to the CDMA cellular system where mobile stations are always moving. Furthermore, with our *Turning Off Redundant Node* and *Select Minimum Neighbor* algorithms, we can achieve an evenly deployed sensor network such that a node normally does not have two neighbors that have very large difference in distance.

The disadvantage of FH-CDMA is that it hard to obtain a high processing gain compared to the DS-CDMA. FH-CDMA also require a very fast switching frequency synthesizer that is hard to build within a sensor node.

Everything comes at a cost. The trade off of using CDMA spread spectrum approach is the increased complexity design at the receiver side. CDMA uses multiple correlating receivers called *rake receiver*. The design of low power DSSS modem within a sensor node is an open research area [8].

4 The Protocol Design

In this section, we elaborate the network formation process in CSMAC. Our network formation process has several phases as illustrated by Figure 2.

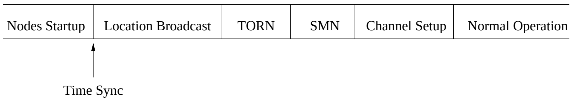

Figure 2: *Network Formation Phases*

We make the following assumptions in our protocol design:

- Each node starts up at almost same time.
- Nodes need not be synchronized at start up, but should be eventually synchronized before the beginning of location broadcast phase. The granularity of time synchronization required is low for our scheme (e.g., one second is enough). In this paper, we assume that each node in the network can be synchronized to a system clock. The timing reference can be provided by one or more GPS enabled nodes or use some technique such as the algorithm proposed by Lamport [15]. As CSMA is employed during the network formation stage, the time frame of each phase should be long enough to provide each node with an opportunity to transmit.
- We assume that each node can estimate its location. This location information can be relative to a reference and does not need to be the absolute location. Sensors can also use either internal GPS receiver or alternate location estimation approaches [2]

4.1 Location Broadcast

In the location broadcast phase, each node uses a CSMA/CA based approach to broadcast its location information to its radio range neighbors. As this is a broadcast, nodes cannot use RTS/CTS.

To reduce the probability of collision and make sure each node has an opportunity for a successful broadcast, we employ large contention window sizes and let each node transmit its location information several times. By using CSMA/CA, we can not eliminate the hidden node problem shown in Figure 3. Because there is an obstacle between A and B, they can not detect each other. A and B may transmit at

---

[2] Please refer Bulusu [7] for a survey of location techniques for sensor networks.

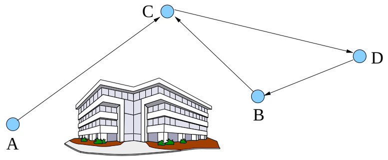

Figure 3: *Hidden Node Problem*

the same time and the signals will collide at node C. Thus C can not record the location information of A and B. As this is a broadcast and there is no negative acknowledgement message from C, both A and B will not become aware of the problem. But when C get the media to transmit, both A and B can hear the message and record the location information of C in their memory. We will show how this hidden node problem can be resolved during the *TORN* and *Channel Setup* phases later.

At the end of the location broadcast phase, each node should have a list with the locations of its radio range neighbors. We call this list *Redundant Neighbor List (RNL)*. We now describe the *TORN* and *SMN* algorithms and then we discuss the channel allocation and network formation process. The backup process of inactive nodes will be presented last.

4.2 Turning Off Redundant Node (TORN)

Once a node has an RNL, it ranks all its neighbors according to the distance between itself and a neighbor. Before deciding who should be selected as its neighbor, the node first decides if there are any neighbors within its redundant range. If sensor nodes are deployed with high redundancy, the probability of having neighbors within redundant range is high.

We denote the sensing resolution as SR (e.g., two meters). Any node within the circle with radius of SR will be treated as a redundant node. We assume redundancy relationship between two (or more) nodes is symmetric, e.g., if node A thinks node B is in its redundant range, node B also thinks node A is in its redundant range.

Similarly, each node uses a CSMA/CA based approach to negotiate who should keep active and who should turn off. By doing this, a random timer is used to avoid collisions. The first node that get the media to transmit can inform its redundant neighbor(s) to turn off by including the ID numbers of these nodes. When a node receives a message asking it to turn off, it will record the ID number of the node issuing the turn off request. It then converts to be a backup node to the issuing node. At the same time, other nodes that receive the turn off message will remove the entries of those nodes requested to turn off.

For a node requested to turn off, it keeps alive until it gets the *receiving frequency* (more about this later) of the node it is backing up until the *Channel Setup* phase. The inactive node then goes to sleep and wakes up only after some period of time to check the energy level of the active node and decide whether it should take over the responsibility of the active node. We discuss backup process in subsection 4.5.

An example is given in Figure 4. We assume the radio range to be such that each node can reach every other node within this small area. Suppose node B is within the sensing resolution (SR) range of node A and B is also within the SR range of node C. Note both A and C are within the SR range of node B. Suppose they have RNL as follows (the redundant nodes are in ***bold italic*** font):

- A RNL: ***B***, C, D, M, H, E, O, . . .
- B RNL: ***A, C,*** D, H, G, E, M, . . .
- C RNL: ***B***, H, A, E, G, F, D, . . .

If node A grabs the media to transmit first, it will transmit a turn off message including the ID number of node B. When node B receives this message, it turns itself into a backup node to node A and broadcasts that it will become an inactive node. When node C receives this turning off message from A to B or the broadcast message of B, C knows that B will become an inactive node and C will remove node B from its RNL. If node B gets the media to transmit before A and C, it will transmit a turn off message including the ID numbers of both node A and node C. Node B can randomly select one node as its primary backup node and the other as second. When other nodes receive this turning off message (such as D, H, etc.), they will remove both A and B from their RNL. If we do not want a node to have too many backups, we can set the expire timer of a node based on the number of redundant neighbors. For example,

```
Timer = (Random delay) + (Constant delay) *
           (Number of redundant neighbors - 1)
```

If we set the *Constant delay* to be larger than the contention window size, a node that has more redundant neighbors will be less likely to become an active node during TORN phase. We do this to provide fairness. For example, if B becomes the active node, both A and C will turn off. While B has two backups, M, D, E, and H do not have even one.

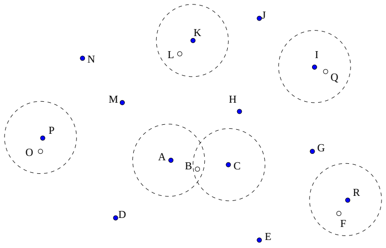

Figure 4: *A TORN Example*

The goal of TORN is to preserve the energy of redundant nodes for future use and prolong the lifetime of the whole sensor network. In practical deployment, we can bundle two (or more) sensors together (if the deployment can not be planned) or deliberately put two (or more) sensors within the SR range (if the deployment can be planned), our TORN algorithm will turn one (or more) redundant nodes into inactive nodes. These inactive nodes only turn on when those active nodes are near depletion of their energy or physically damaged.

At the end of TORN phase, only active nodes are left in the network. The resulting neighbor list in each active node is called *non-redundant neighbor list (NNL)*. This NNL will be used in the SMN process, which we discuss in the next subsection.

Hidden Node Problem: Consider Figure 3. Assume B is within the SR range of C, but C does not know B's location. So C thinks it does not have a redundant neighbor and keeps silent during the TORN phase. But because B has the location information of C, it knows that C is its redundant node. When B grabs the media, it sends an RTS request including C's ID number and its own ID number. When C receives this RTS message, it replies with a CTS message including the ID numbers of both B and C. B then sends a turn off request to C with its (B) location information. C then knows that it has a redundant neighbor B, and perhaps due to a hidden node problem or collision, it did not record B's location in its RNL at location broadcast phase. C broadcasts a sleep message to inform its neighbors that it will become an inactive node. When A receives this message, it removes C from its RNL.

4.3 Select Minimum Neighbor (SMN)

After the TORN stage, each active node has a list (NNL) of location information of its radio range neighbors. A node will not select all of them as neighbors. It only selects a node as its neighbor if *there is no other neighbor that can provide a multi-hop path with lower power consumption*. We designed an algorithm for a sensor node (we call it *seed node*) to select its neighbors. With the list (NNL) of active neighbors, the seed node ranks them according to their distances to the seed. Then starting from the last node (furthermost) in NNL, each node will be tested against other neighbors in the NNL based on a specific radio propagation model (e.g., two-ray ground) and an electronic energy dissipation model. If any neighbors in the NNL can provide a multi-hop path from seed to the current tested node with less power consumption, the current tested node will not be selected as a neighbor and will be removed from NNL. The algorithm is shown below:

Select Minimum Neighbor Algorithm:

```
INPUT: A list of neighbor nodes with its location
       information (ranked according to distance
       to seed node)
OUPUT: A list contains minimum number of neighbors

BEGIN:
 VECTOR INPUT = List of neighbor nodes
 VECTOR OUTPUT = NULL;
```

```
 NODE SEED = seed node;
 NODE CURRENT = NULL;
 NODE TEMP = NULL;
 BOOLEAN isNeighbor = TRUE;
 DOUBLE p1, p2, p3; // Transmission power
 DOUBLE Tx_elec; // Tx electronic power consumption
 DOUBLE Rx_elec; // Rx electronic power consumption

 LOOP1:
  CURRENT = last node in INPUT vector;
  p1 = Tx power required from SEED to CURRENT;
  isNeighbor = TRUE;

  IF (INPUT IS NOT NULL)
   LOOP2:
    TEMP = next node in INPUT;
    p2 = Tx power required from SEED to TEMP;
    p3 = Tx power required from TEMP to CURRENT;
    IF (p1 > p2 + p3 + Tx_elec + Rx_elec)
    THEN { isNeighbor = FALSE; BREAK LOOP2; }
   END OF LOOP2

  IF (isNeighbor IS TRUE)
  THEN {Add CURRENT to OUTPUT vector;}
  REMOVE CURRENT from INPUT vector;
 END OF LOOP1

 RETURN OUTPUT;
END
```

After the SMN phase, each active node only has a minimum set of neighbors and we call the resulting neighbor list *minimum neighbor list (MNL)*. This MNL will be used in the *Channel Setup* process wherein a peer-to-peer communication channel will be setup for each neighbor in MNL and the seed.

By using *TORN* and *SMN*, we want to achieve an evenly deployed sensor network that is energy-efficient without sacrificing the sensing accuracy. Figure 5 shows the point of view of an active sensor node after the TORN and SMN phases. $R1$ denotes the sensing resolution range and $R2$ denotes the radio transmission range. Node A has one redundant node and only selects five nodes as its neighbors although its radio can reach more nodes.

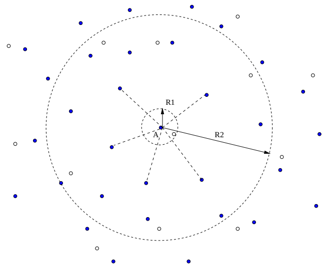

Figure 5: *An Active Node's Point of View after TORN and SMN phases.*

4.4 Channel Allocation and Network Formation

At the end of SMN phase, each node only has minimum set of neighbors. The last step is to allow each node to setup connections to all it neighbors in the MNL.

4.4.1 Channel Allocation Pattern

Before we discuss the channel allocation process, we first specify the channel allocation pattern. If we need to use orthogonal PN codes, we only have limited codes available. Assume we use 128 chip/bit

Walsh code sets, we only have 128 codes available instead of 2^{128}. This will cause interference problems if all nodes use the same frequency. As shown in Figure 6 (a), which illustrates A and B negotiating to use PN1 for communication from A to B. We can avoid both node A and node B using PN1 again with their neighbors but we can not avoid node C and node D using the same code. The conditional probability of this scenario is $P((CD = PN1)|(AB = PN1)) = P((AB = PN1)(CD = PN1))/P(AB = PN1) = 1/128$, which is not very low. Even if we use stringent power control, the interference may still exist. We propose two approaches to resolve this problem, namely using *spatial diversity* and *frequency diversity*.

By *spatial diversity*, we separate nodes using the same PN code spatially. We make each node negotiate with its neighbors using full transmission power so that each radio range neighbor can hear these messages. If node C and node D are within the radio range of node A or node B, C and D can hear the negotiations between A and B. Thus they know that PN1 was used by other nodes located within their radio range. This way node C and node D will not select PN1 for communication. When the network enters normal operation phase, each node will no longer transmit at full power but only communicate with its minimum neighbor with much lower calibrated transmission power.

A problem with this approach is that if the radio range can cover more than 64 (128 PN codes can only serve 64 node pairs) nodes, some nodes may not be able to get a PN code to use. This approach has direct relation to the density of sensor network. The density can range from few sensor nodes to hundred sensor nodes in a region, which can be less than 10m in diameter. The density μ is calculated as [7]:

$$\mu(R) = \frac{N\pi R^2}{A} \tag{6}$$

where N is the scattered sensor nodes in region A, and R is the radio transmission range. Generally, $\mu(R)$ gives the number of nodes within the transmission radius of each node in region A. To make this approach functional, the density μ must be less than half the available PN codes. Two major problems with this approach are: 1) Tx and Rx can not happen simultaneously because the transmitter power is so high that it will drown out any receive messages, 2) CDMA *near-far* problem.

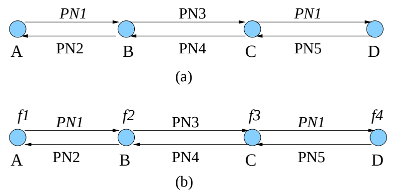

Figure 6: *Channel Allocation Patterns*

In *frequency diversity* approach, we use FDMA plus CDMA as shown in Figure 6 (b). Each node uses a different frequency band to *receive* signals. As bandwidth is not a scare resource in sensor networks, this may be a practical approach. For example, assume the sensor base band signal is 10Kbps and we use 128 chip/bit. The spreading bandwidth is 1.28MHz. If we use 902-928 Industry, Scientific, and Medical (ISM) band, we can have about 20 frequency bands. This will give us $128 \times 20 = 2560$ channels! If we use a higher frequency spectrum, such as 2-3GHz, we can have more frequency channels. The probability of two nodes selecting both the same frequency band and the same PN code is very low. By using frequency diversity, both the Tx and Rx can happen simultaneously and the interference caused by competing transmissions at a specific receiving node are also reduced.

In CDMA, all received signals are noises except the intended signal if the same frequency is used. Once this noise floor reaches a certain level, the intended signal is unable to be retrieved. If the receiver of each node can use a different receiving frequency, those transmission signals from non-neighbors will not be able to penetrate the band pass filter at the front-end of the receiver. Thus those signals will not contribute to the noise floor. Assume a sensor node is receiving from a neighbor, those nodes that may contribute to the noise floor are the ones listed in its MNL. And those noise, if existed, are normally expected signals from its other neighbors. The problem with this approach is that the transmitter is required to synthesized to different frequencies for transmission to different neighbors and thus this approach is not suitable for broadcast (we are currently designing a more efficient broadcasting scheme with frequency diversity). We adopted frequency diversity in our design and simulation because of its better latency performance versus spatial diversity. We plan to incorporate spatial diversity in future research because it has better broadcast performance and simpler transceiver design, especially when ultra-wide-band (UWB) technology becomes available for sensor networks.

4.4.2 Channel Allocation Process

We now describe the channel allocation process. At this stage, each node no longer transmits at full power. Each node estimates the transmission power required to reach its furthermost neighbor in its MNL and uses this power level for the negotiation process. This allows other nodes that are far enough from this node to initiate another setup process simultaneously. Similarly, CSMA/CA is used by nodes to setup connections with each other. When a node grabs the media, it will hold the media until it finishes the channel allocation with *all* of its neighbors in the MNL.

Figure 4 illustrates this process. At the beginning of *Channel Setup* phase, each node sets a random timer and begins to count down.

1. Assume node A's random timer expires first, A sends a Request-to-Send (RTS) packet to its first neighbor in the MNL, in this case, C.
2. If C is not in another channel setup process with one of its other neighbors, say G, C will respond with a Clear-to-Send (CTS) packet. Otherwise, C will not respond the request of node A.
3. If A does not receive the CTS packet from C in time, A knows that C is busy with another channel setup process and will back-off its initiation. Other nodes that receive the RTS or CTS packet will hold active initiation for channel setup and only passively reply if needed.
4. Suppose C is not in another channel setup process, when A receives the CTS packet from C, A will initiate a channel setup packet to C. We call this packet SYN packet. There are two possible cases for A:
   (a) A is not connected with any other neighbor (in this case, the receiving frequency of A is not fixed) and,
   (b) A is connected with some other neighbors (in this case, the receiving frequency of A is fixed).

   In the first case, A will randomly select a frequency band as its receiving frequency and a random PN code as its transmission PN code to node C. In the second case, the receiving frequency of A is fixed and it only randomly selects a PN code to node C. A must guarantee that this PN code has no conflict with PN codes employed by A for communicating with its other neighbors. A then encapsulates its receiving frequency, a flag to indicate whether this receiving frequency is fixed, its transmitting PN code to C, and the ID number of A and C into a SYN packet and transmit to C.
5. When C receives this SYN packet, it will reply with a SYNACK or SYNNAK packet. There are also two possible cases for node C:
   (a) C is not connected with any other neighbors (In this case, the receiving frequency of C is not fixed) and,

(b) C is already connected with some other neighbors (In this case, the receiving frequency of C is fixed).

In the first case, C will select a random receiving frequency and a random transmission PN code that differs from the receiving frequency and PN code of A. In the second case, C first checks if there is any conflict of the transmission PN code and receiving frequency of A (If receiving frequency flag of A is not set) with its own receiving frequency and the PN codes employed with its other neighbors. If any conflict occurs, C will reply with a negative acknowledgment packet (SYNNAK) indicating that the transmission PN code or receiving frequency of A is invalid. C then randomly selects its receiving frequency and transmission PN code in the first case and only randomly select a transmission PN code in the second case. The PN code selected by C must meet following criteria: 1) no conflict with the PN codes that it employed with its other neighbors, 2) if C and A happen to employ the same receiving frequency and both their receiving frequencies are fixed (this may happen when both A and C are attached to other neighbors before their own channel setup process although this probability is low), C must guarantee that the PN codes employed are different. C then encapsulates its validation of the transmission PN code and receiving frequency of A, its own receiving frequency, a flag to indicate whether its receiving frequency is fixed, its transmitting PN code to A, and the ID number of C and A into a SYNACK packet and transmits to A.

6. When A receives the SYNACK (or SYNNAK) packet from C, it first checks whether the acknowledgement is positive or negative. If the acknowledgement is negative, A will re-select the PN code or receiving frequency (if not fixed) and initiate a new SYN packet to C. If the acknowledgment is positive, A uses a similar validation process to validate the transmission PN code and the receiving frequency of C. If any conflict occurs, A will reply with a negative acknowledgment (NAK) to C.
7. When C receives a NAK packet, C will re-select its PN code or receiving frequency and reply with a new SYNACK packet to A.
8. If no conflict occurs, A will reply with a positive acknowledgment (ACK) packet to C.
9. When C receives this ACK packet, it knows that the PN code and receiving frequency allocation has succeeded.
10. After this, A and C may calculate the required transmission power to each other based on the distance and possible channel characteristics (e.g., SNR, BER, RSSI, etc.). A and C then synthesize their receivers to their respective receiving frequencies and conduct a pilot run that should be initiated by A. A and C may exchange several small packets to setup their transmission power to the correct level. When the pilot run finishes, the channel setup between A and C is finished. A and C then synthesize back to CSMA mode for further connection setup process. A will repeat this process until it finishes the connections with all of its other neighbors. After that, A will broadcast a clear packet indicating that it finished the connection setup and the media is now released. When other nodes receive a clear packet, they can initiate connection setup process with their neighbors if needed.

The packets exchanged between nodes during the connection setup are:

- RTS/CTS/Clear, control packets used to access the media;
- SYN, sent by the active node to initiate a channel setup process;
- SYNACK/SYNNAK, response to SYN packet, send by the passive node that is required to start a channel setup process;
- ACK/NAK, acknowledgment to the SYNACK packet, either positive or negative.

Figure 7 shows three different scenarios in the channel setup process. where (a) represents the normal operation, (b) shows the failed validation of PN code or receiving frequency of A, and (c) shows the failed validation of PN code or receiving frequency of C. After the channel setup phase, each node has the listed information stored in its memory:

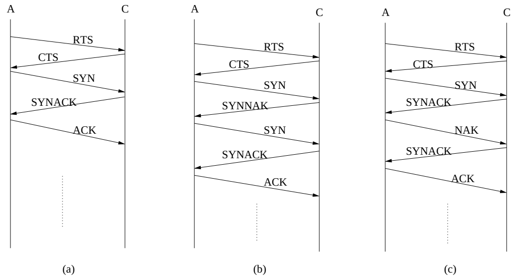

Figure 7: *(a) Normal Operation. (b) C disagrees with the PN code or Receiving Frequency of A. (c) A disagrees with the PN code or Receiving Frequency of C.*

```
Node Information:
  Rx Frequency -- the receiving frequency
                  of this node

Each Neighbor Information:
  ID -- the neighbor identity number
  Rx PN code -- from neighbor to this node
  Tx PN code -- from this node to neighbor
  Tx frequency -- the receiving frequency
                  of neighbor
  Tx power level -- transmission power level
                    from this node to neighbor
```

4.5 Backup Process

In this subsection, we provide a brief description about the backup process (as we are working on a detailed backup process at this stage). After the *channel setup* phase, an inactive node turns off *completely* and wakes up after some period of time. The inactive node then checks the energy level of the active node that it is backing up and decides whether it should take over the responsibilities of the active node. The checking period is based on the energy level of the active node. E.g., at the beginning of the deployment of sensor network, the active node has full energy level, the checking period can be long, such as once a day. When the energy level of the active node drops to a certain level, the checking period should be decreased, such as once an hour. When the energy of the active node is near depletion, the backup node and active node negotiate with each other and transfer the responsibility to the backup node.

4.6 Summary

We described a novel CDMA-based, self-organizing, location-aware MAC protocol for ad hoc wireless sensor networks. Our network formation process comprises of several phases including *Location Broadcast, Turn off Redundant Node (TORN), Select Minimum neighbor (SMN), Channel Setup, and Normal Operation.* The TORN scheme is employed to preserve the energies of redundant nodes for future use while still keep sufficient sensing accuracy. The SMN algorithm is employed by a sensor node to select a minimal subset of communicating neighbors to conserve energy. CDMA approach is used for efficient low latency performance by allowing simultaneous transmissions.

We have not designed a sleep and wakeup scheme at this stage. Part of the reasons is that we are targeting the applications that have higher traffic and stringent latency requirements. More discussions about sleep and wakeup are presented in 6.4.

5 Simulations and Analysis

We implemented and simulated our protocol in Network Simulator (NS-2). All phases except *Turning off Redundant Nodes (TORN)* have been finished so far. These phases include *location broadcast, select minimum neighbor (SMN), channel setup, and normal operation*. DSSS (Directed Sequence Spread Spectrum) is simulated as a PN code attribute in packet header. When the packet is received at a receiving node, this PN code is checked against PN codes the receiving node is monitoring. If a match is found, the packet is passed to the next step for further processing. If no match is found, the packet is discarded. This procedure is used to simulate the de-spreading process. In this simulation, we call our protocol CSMAC (denote CDMA Sensor MAC).

The goals of this simulation are to measure the energy efficiency and latency trade-offs in CSMAC. The idle energy consumption is not traced at this stage (the idle energy consumption and sleep mode is further discussed in 6.4). Also, the energy consumed at the setup phase of CSMAC is not traced. As a comparison, we also measured the performance of SMAC [12] as SMAC was also implemented in NS-2.

SMAC is similar to IEEE 802.11 but uses a periodic sleep to save idle energy consumption and turns off nodes during data transmission by neighbors to save the energy consumed in overhearing.

We adopted a simple energy dissipation model that is used in LEACH [11]. The model is shown in Figure 8. This model separates the electronic energy consumption and amplifier energy consumption at the transmitter side. For comparison purposes, we used the same physical parameters for both CSMAC and SMAC.

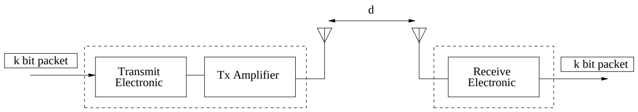

Figure 8: *Radio energy dissipation model*

```
Data Rate: 10Kbps
Propagation Model: TwoRayGround
Antenna Height: 0.1 meter
Antenna Gain: 1
System Loss: l
Rx Threshold: 1nW
Carrier Sense Threshold: 0.1nW
Rx Elec Power: 20mW
Tx Elec Power: 20mW
Full Radio Signal Power: 200mW
ISM Frequency Band: 902-928 MHz
```

More detailed physical parameters are not implemented at this stage. For example, the efficiency of the amplifier in general, is only about 30∼40%. CDMA consumes more energy for its coding and decoding. But on the contrary, CDMA provides a better channel, and as a result, can use lower transmission power to achieve the same bit-error-rate (BER) performance. We plan to implement these in our future simulations.

We use two simple topologies that were used by SMAC to do the measurements.

5.1 Measurement with Two-Hop Network Topology

CSMAC uses contentionless approach and the communications are peer-to-peer. Broadcasting means multi-unicast dependent on the number of neighbors (a more efficient broadcast implementation is still under progress). This means CSMAC is not efficient with broadcast traffic but on the contrary, is efficient with unicast traffic. We tested both (i) the pure broadcast traffic and (ii) the combined broadcast and unicast traffic with a simple topology shown in figure 9. The routing protocol used in this simulation is DSDV (Destination Sequence Distance Vector) and application used is CBR (Constant Bit Rate) traffic. The radio signal power is set to 20mW. The distances between nodes are deliberately set to make sure nodes 0, 1, 3 can hear each other and nodes 2, 1, 4 can hear each other.

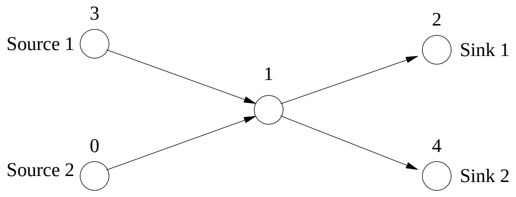

Figure 9: *Two-hop network with two pairs of sources and sinks*

To test pure broadcast traffic, we simply did not generate the application traffic. DSDV routing protocol uses broadcast to exchange routing information periodically with its neighbors. Figure 10 shows the average node energy consumption with pure broadcast traffic. The figure shows that CSMAC consumes 50% higher mean energy per node than SMAC. We then tested a scenario with combined broadcast and unicast traffic. Two sources and two sinks are used. The CBR traffic interval is set to 5 seconds. Figure 11 shows the average node energy consumption with combined broadcast and unicast traffic. Surprisingly, CSMAC consumes 27% less mean energy per node because the CBR traffic (unicast) contributes to the major part of the energy consumed. CSMAC also eliminates RTS, CTS, and ACK packet exchange that are essentials in SMAC. In addition, CSMAC achieves a 62% lower mean packet latency as shown in Figures 12 and 13. In most cases, the latency using CSMAC is simply the accumulation of the multi-hop transmission time. The first packet latency is much higher than the others because at the beginning, each node has to use ARP (Address Resolution Protocol) to resolve the MAC address of next hop before it can forward the data packet. Following packets do not need ARP anymore. This is the same for both CSMAC and SMAC.

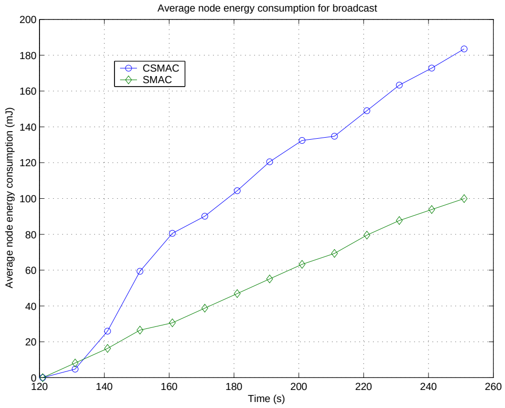

Figure 10: *Average per node energy consumption for broadcast traffic*

We also tested this topology with *directed diffusion* routing protocol. Directed diffusion is a routing protocol especially suitable for distributed sensor networks. Directed diffusion was implemented in NS-2 and a ping application is provided for testing purposes (for more information, please refer to NS-2 manual).

We used two different ping senders and ping receivers to simulate two sinks and two sources. The measurement time is from 131s to 851s. Totally 73 ping messages have been sent between

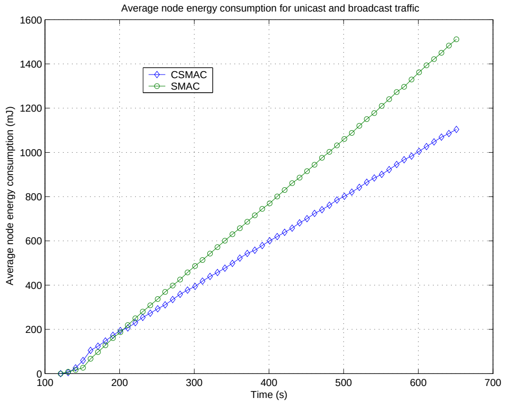

Figure 11: *Average per node energy consumption for both unicast and broadcast traffic*

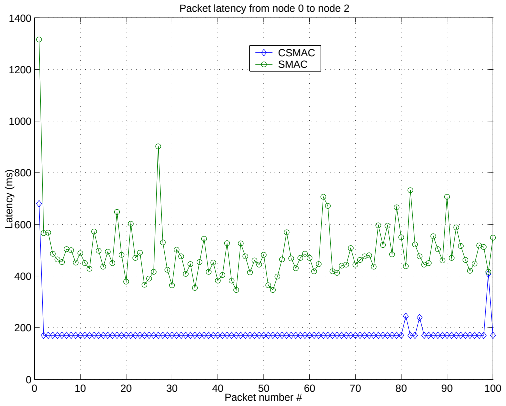

Figure 12: *Packet latency from node 0 to node 2*

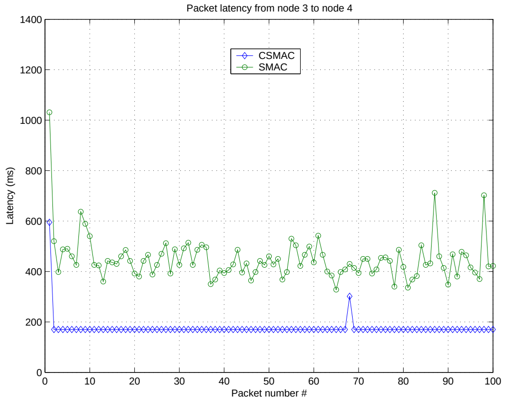

Figure 13: *Packet latency from node 3 to node 4*

a source and sink pair. The ping message interval is set to 10 seconds. Figures 14 shows that CSMAC consumes 34% lower mean energy per node compared to SMAC. Figures 15 and 16 show the latency performance. CSMAC shows 59% lower mean message latency from node 0 to node 2 and 72% lower mean message latency from node 3 to node 4 compares to SMAC.

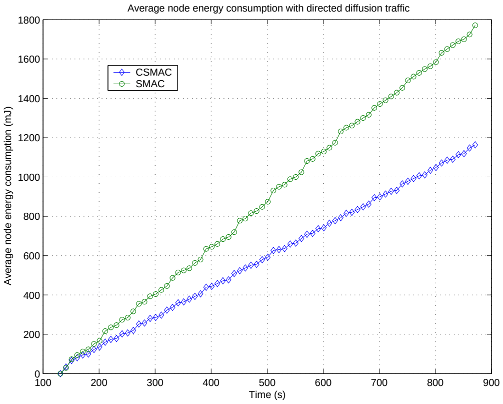

Figure 14: *Average node energy consumption with directed diffusion traffic*

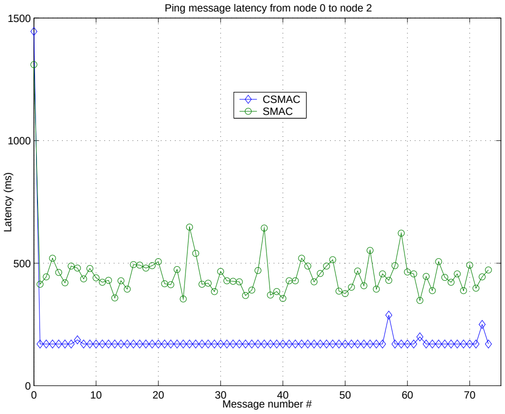

Figure 15: *Ping message latency from node 0 to node 2*

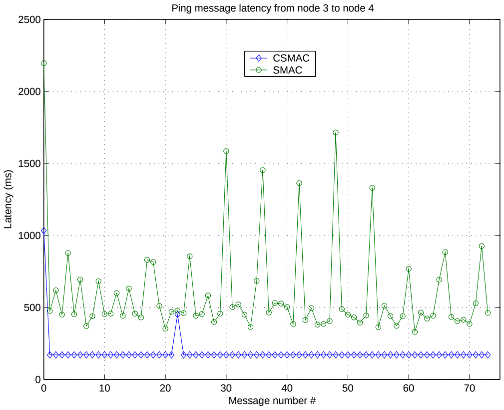

Figure 16: *Ping message latency from node 3 to node 4*

5.2 Measurement with Ten-Hop Network Topology

We next tested the energy consumption and latency performance with a linear ten-hop network topology as shown in Figure 17.

We tested by using directed diffusion protocol and the ping application. Only one sink and one source are used in this case. Figure 18 shows that CSMAC consumes 41% lower mean energy per node. This figure shows that CSMAC achieves better performance than for a two-

Figure 17: *Ten-hop network with one source and one sink*

hop topology because broadcast traffic is not severe in this linear topology. To simulate the CDMA may consume more power because of encoding and decoding process, we increase the electronic power consumption to 40mW (both Rx and Tx). After this increase, CSMAC achieves the similar average node energy consumption versus SMAC. This means that if the electronic energy consumption for CDMA codec can achieve less than two times of normal non-CDMA electronics, we can achieve the similar energy consumption in this linear topology (do not forget that CDMA can provide a much better channel!). But CSMAC achieves 74% lower mean latency as shown in Figure 19.

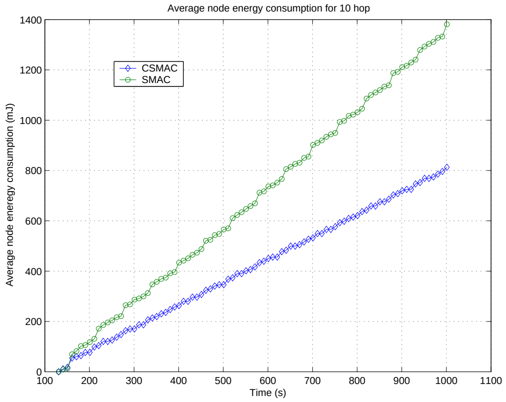

Figure 18: *Average node energy consumption for 10 hops*

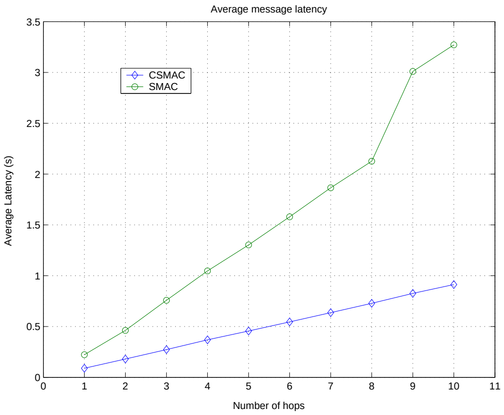

Figure 19: *Average message latency for 10 hops*

5.3 Measurement of SMN with Random Topology

In this subsection, we tested our *select minimum neighbor (SMN)* algorithm with 20 randomly deployed sensor nodes in a 50m × 50m area. The initial radio signal power is set to 200mW, which equals the full radio signal power. This is to ensure that each node can reach all its neighbors within its radio range. After the channel setup phase, the radio power is recalculated according to the distance a node is from its neighbor.

Figure 20 shows the node connection pattern before SMN and Figure 21 shows the node connection pattern after SMN. From the diagrams we can see that after SMN, some nodes removed part of ther neighbors to conserve energy. For example, nodes 6 and 8 are neighbors before SMN but are not neighbors after SMN. This is the same for nodes 6 and 4, nodes 2 and 18, nodes 7 and 19, nodes 10 and 14, etc.

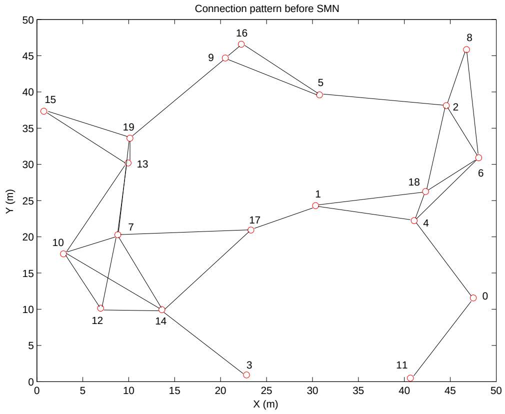

Figure 20: *Node connection pattern before using SMN*

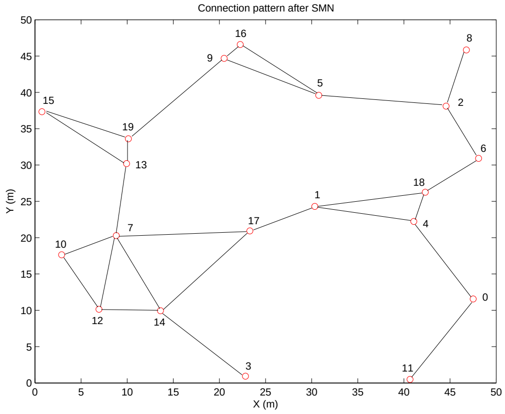

Figure 21: *Node connection pattern after using SMN*

We tested the energy consumption and latency by using two pairs of sources and sinks with directed diffusion ping application. One pair of source and sink is put on node 15 and node 0, the other is put on node 8 and node 10. The simulation time is from 241s to 551s. The ping message interval is 10 seconds. Totally 31 ping messages have been received at each sink.

Figure 22 shows the comparison of energy consumption with and without SMN. The diagram shows that 21% less mean energy is consumed with SMN. Figures 23 and 24 show the message latency comparison. From the diagram we can see that the latency from node 15 to node 0 does not have much difference. This is because the minimum number of hops from node 15 to node 0 did not change after SMN. But for the message latency from node 8 to node 10, non-SMN achieves the better latency performance because non-SMN topology can provide a route with less number of hops compared to SMN topology. This reveals the trade-offs between the energy consumption and latency performance.

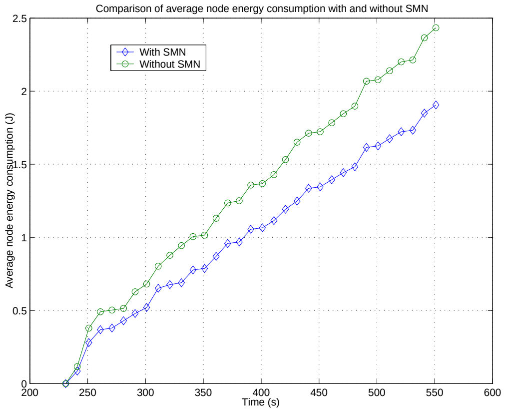

Figure 22: *Comparison of average energy consumption with and without SMN*

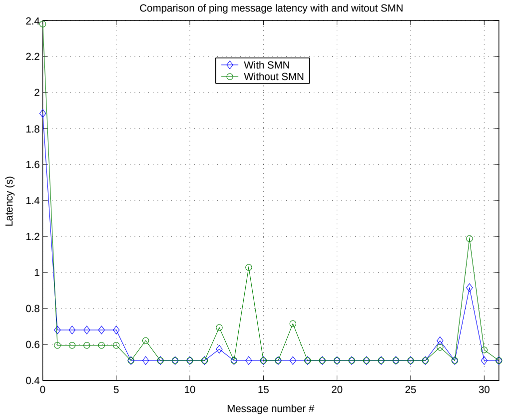

Figure 23: *Comparison of message latency with and without SMN from node 15 to node 0*

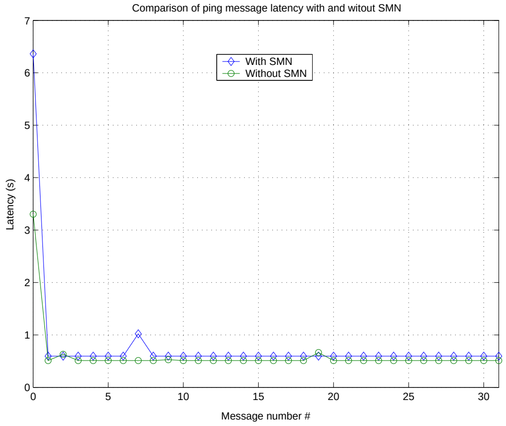

Figure 24: *Comparison of message latency with and without SMN from node 8 to node 10*

6 Future Research

In this section, we comment on further improvements to our protocol. We are currently working on some of them.

6.1 Broadcasting with Frequency Diversity

Broadcasting in our current design is implemented by multiple unicast to all neighbors, which is not energy-efficient. An improved broadcast implementation is to employ two receivers in a sensor node. One receiver synthesizes to a dedicated frequency band and is used for broadcast traffic. This dedicated frequency band is common across all sensor nodes. A dedicated PN code can be allocated by a sensor node for its broadcast data to all its neighbors. The detailed scheme is still under design.

6.2 Ultra-Wide-Band (UWB) with Spatial Diversity

We mentioned in section 4.4 that ultra-wide-band (UWB) is an attractive candidate technique for sensor networks. UWB employs baseband transmission and thus requires no intermediate or radio carrier frequencies. UWB has several advantages such as resilience to multi-path, low transmission power, and simple transceiver circuitry design. In general, UWB uses pulse position modulation (PPM). Because UWB employs baseband transmission without carrier frequencies, the frequency diversity we proposed in Section 4.4 can not be used and spatial diversity is the choice with UWB. We believe this is an interesting and attractive direction and worth exhaustive research efforts.

6.3 SMN Improvements

The SMN algorithm we described in Section 4.3 is only based on a specific radio propagation model (e.g. two-ray ground), which is not enough to reflect the real characteristics of a channel. In practice, the algorithm should also take into account other physical parameters such as SNR (Signal-to-Noise Ration), BER (Bit-Error-Rate), RSSI (Received Signal Strength Indicator), etc. And we believe the current version of SMN algorithm is not an optimized solution for neighbor selection. For example, current SMN algorithm only checks one extra hop to a specific neighbor. While it is possible that a two-hop route may not save energy, three or more hops may. Another issue is that even though a multi-hop route can provide energy savings, it also increases the packet latency. If the energy saved is not very significant, it may not be worthwhile to incur a large latency overhead. Balancing the energy consumption and latency plus aforementioned physical parameters (SNR, BER, RSSI) is an optimization problem. A more efficient SMN algorithm is still under design.

6.4 Application-driven Duty Cycles

We have not designed a sleep and wakeup algorithm at this stage because we are targeting the applications that have higher traffic and stringent latency requirements. Sleep mode saves energy consumed by radios in idle time. Although this approach seemingly provides significant energy gains, it is important to note that sensor nodes communicate using short data packets. The shorter the packets, the more dominant the startup energy costs. If we blindly turn the radio off during each idle slot, over a period of time we might end up expending more energy than if the radio had been left on [1]. This is especially so when traffic is higher (when an event occurs) or when the sensor network is required for real time information transfer to information consumers.

If the traffic is indeed low and applications can tolerate long latency, we can adopt some alternate sleep and wakeup schemes such as SMAC's periodic sleep and wakeup approach. Guo et al [14] proposed a super low power radio called *wake-up radio* that can allow the receiver to power down during idle listening time. The *wake-up radio* serves as a *small ear* and keeps monitoring the channel signal on a super low power. The monitoring power can be less than $10\mu W$. If a node needs to send a packet out, it simply wakes up. If a neighbor needs to send a packet to this node, it will send a short *wakeup beacon* using the wake-up radio channel. Upon reception of this beacon the wakeup radio will trigger the power up of the normal radio to be ready for reception.

Another approach to save idle energy is to allow sensors to sleep during non-duty cycles based on opportunistic application dependent criteria (e.g., no monitoring during night time) rather than simply turning sensors on and off based on redundant density criteria. For example, in the bush fire monitoring application, we only need sensors to be on duty during day time. The sensors can sleep during night time as normally bush fires seldom occur without sunshine (exclude the deliberate arsonist). If a sensor has an internal clock, the sensor can be set to sleep during night time and wakeup in the morning. We feel that turning sensors on and off based on application level criteria makes more sense than simply turning them off at the MAC layer.

A detailed back up scheme and responsibility transferring process is also essential for the integrated design of our protocol. Further simulations with more detailed physical parameters and possible implementation in hardware is the next big step.

7 Concluding Remarks

No single MAC protocol is suitable for all sensor network applications. This paper proposed a novel CDMA-based, self-organizing, location-aware MAC protocol design suitable for application scenarios with higher traffic and stringent latency requirements. We also discussed several future improvements and research directions.

Previously proposed MAC protocols for sensor networks have prioritized energy-efficiency foremost, ignoring other requirements. By exploiting CDMA-based techniques, self-organization and location-awareness in network formation (through TORN and SMN), our protocol design balanced the performance requirements of sensor networks such as energy efficiency, low latency, sensing accuracy, and fault tolerance.

Preliminary NS-2 simulation based evaluations seem to suggest that a combination of location-awareness at MAC layer to improve energy-efficiency and CDMA-based techniques to reduce interference and improve network capacity may actually provide greater energy-savings as well as much better latency performance in a multi-hop network. We are currently investigating an implementation of our CDMA-based media access protocol to verify the results.

References

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, “A Survey on Sensor Networks”, IEEE Communications Magazine, August 2002, pp. 102-114.

[2] G. J. Pottie and W. J. Kaiser, “Wireless Integrated Network Sensors,” Commun. ACM, vol. 43, no. 5, May 2000, pp. 551-58.

[3] E. Shih et al., “Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,” Proc. ACM MobiCom ’01, Rome, Italy, July 2001, pp. 272-86.

[4] A. Woo, and D. Culler, “A Transmission Control Scheme for Media Access in Sensor Networks,” Proc. ACM MobiCom ’01, Rome, Italy, July 2001, pp.221-35.

[5] K. Sohrabi et al., “Protocols for Self-Organization of a Wireless Sensor Network,” IEEE Pers. Commun., Oct. 2000, pp. 16-27.

[6] K. Sohrabi et al., “Near-Ground Wideband Channel Measurement,” IEEE Proc. VTC, New York, 1999

[7] N. Bulusu, “Self-Configuration Localization Systems,” PhD Dissertation, UCLA, 2002

[8] C. Chien, I. Elgorriaga, and C. McConaghy, “Low-Power Direct-Sequence Spread-Spectrum Modem Architecture For Distributed Wireless Sensor Networks,” ISLPED 01, Huntington Beach, CA, Aug. 2001.

[9] S. Singh, C.S. Raghavendra, “PAMAS: Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks”, ACM Computer Communication Review, Vol. 28, No. 3, pp. 5-26. July 1998.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” IEEE Proc. Hawaii Int’l. Conf. Sys. Sci., Jan. 2000, pp. 1-10.

[11] W. R. Heinzelman “Application-Specific Protocol Architecture for Wireless Networks”, PhD dissertation, MIT, June, 2000.

[12] W. Ye, J. Heidemann, D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”, IEEE Proc. Infocom, June 2002, pp.1567-1576.

[13] S. Tilak, N. B. Abu-Ghazaleh, W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Network Models”, Mobile Computing and Communications Review, Vol.1, No. 2.

[14] C. Guo, L.C. Zhong, J.M. Rabaey “Low Power Distributed MAC for Ad Hoc Sensor Radio Networks”, IEEE Proc. GlobeCom 2001, San Antonio, November 25-29, 2001

[15] L. Lamport. “Time, Clocks, and the Ordering of events in a Distributed System”, Communications of ACM, Vol. 21, Number 7, July 1978

[16] William C. Y. Lee, “Mobile Communications Design Fundamentals”, Second Edition, John Wiley and Sons, Inc, 1993

[17] R. Peterson, R. Ziemer, D. Borth “Introduction to Spread Spectrum Communication”, Prentice Hall, 1995

[18] A. J. Viterbi “CDMA Principles of Spread Spectrum Communication”, Addison Wesley, 1995

[19] T. S. Rappaport “Wireless Communications, Principles and Practice”, Second Edition, Prentice Hall, 2002

[20] Jack P.F. Glas “Non-Cellular Wireless Communication Systems,”, PhD Thesis, Electrical Engineering, Delft University of Technology, 1996