A MAC-Layer Retransmission Algorithm Designed for the Physical-Layer Characteristics of Clustered Sensor Networks

Qingjiang Tian and Edward J. Coyle
Center for Wireless Systems and Applications
School of Electrical and Computer Engineering, Purdue University
465 Northwestern Avenue, West Lafayette IN 47907-2035
{tianq, coyle}@purdue.edu

Abstract

We consider a cluster of sensors that are interrogated by a mobile node that is passing by or loitering overhead. The mobile node functions as a clusterhead that collects one packet of sensed data from each sensor node. A broadcast from the mobile node is used to synchronize the motes in its cluster to within 1 microsecond. We exploit this physical layer characteristic to improve the retransmission algorithm in CSMA/CA-based MAC protocols. The result is a Synchronized, One-Stage-Backoff Retransmission Algorithm (SOSBRA) that is easy to implement and is shown by analysis and simulation to have fewer collisions than binary-exponential-backoff retransmission algorithms. We show via simulation that a SOSBRA-based 802.11 DCF protocol performs significantly better - in terms of energy usage, delay, and throughput - than both 802.15.4 (ZigBee) and 802.11 DCF (WiFi). A comparison between SOSBRA- and TDMA-based MAC protocols is also provided that shows SOSBRA is fair and more fault-tolerant.

1. INTRODUCTION

A common scenario in sensor networks is the collection of information from a large number of wireless sensor nodes by a mobile node that is passing by or loitering overhead. The mobile node's task is to quickly gather one packet of data from each sensor and then move on.

Two of the most critical performance measures for this scenario are: (i) the time required to gather one packet of data from each sensor node; and (ii) the energy expended to communication the data by the sensor nodes. Using these two measures, we compare the performance of several standard approaches to the design of the MAC layer for this scenario: TDMA, 802.11b, and Zigbee. We then demonstrate how the performance of 802.11b and ZigBee can be significantly improved by a new Synchronized One-Stage Backoff Retransmission Algorithm (SOSBRA).

The following assumptions about this scenario are used in the remainder of this paper to:

- The sensor nodes maintain RF silence until the mobile node arrives and announces its presence. This enables them to remain hidden while they gather data but prevents them from exchanging data with each other. The data gathered by the sensors include one or more samples of a random field - the goal will be to gather enough samples to estimate the parameters of that field.
- Each sensor node knows the number of sensor nodes that were deployed. This number can be downloaded to each node before they are deployed.
- Each sensor node in the network has sufficient communication power to transmit directly to the mobile node. The network can thus be modeled as a single cluster [1-11] of nodes with the mobile node as a clusterhead.
- When a node successfully gains control of the channel, it transmits all of the data it collected and processed in a single packet. This packet is followed immediately by an ACK from the mobile node, after which the sensor node returns to gathering and processing data until the arrival of the next request to upload data.
- The sensor nodes are within X hundred meters of the mobile node. The transmission from the mobile node that is used to initiate the data collection can be used to synchronize the sensor nodes' clocks to within X microseconds. This synchronization task can be repeated by each ACK and is similar to the synchronization beacon used in the ZigBee (IEEE 802.15.4) protocol [12].

In this paper, we also employ a synchronization beacon to synchronize the cluster. Unlike ZigBee, though, we also use the beacon to distribute information to motes that enables them to optimally tune any algorithm they use to retransmit packets involved in collisions in the MAC layer. We redesign the retransmission algorithms to extract maximum benefit from this information. The new algorithm replaces the binary-exponential-backoff algorithm used in current CSMA/CA-based MAC algorithms with a new Synchronized, One-Stage-Backoff Retransmission Algorithm (SOSBRA). The information that is then broadcast in the synchronization beacon is the backoff-window size that all motes in the cluster will use to schedule their retransmissions. When a collision does occur, its retransmission is scheduled in the next backoff window.

This new approach to managing retransmissions in sensor networks is simpler than others and offers better performance. To demonstrate the improved performance, we find the density function of the time to gather one packet from each node in a cluster with a SOSBRA-based 802.11 DCF protocol. Numerical and simulation results show that this new protocol outperforms both WiFi (802.11 DCF) [13] and ZigBee (802.15.4). It results in fewer collisions and thus uses less energy and wastes less time on the channel. We also compare this new approach to TDMA, which requires the motes to be ordered as well as synchronized, and show that the random access approach requires less overhead and provides greater flexibility.

Paper organization: Related work is reviewed in Section 2. In Section 3, we investigate the performance of 802.11 DCF in a clustered environment and propose a new CSMA/CA protocol with SOSBRA, the synchronized one-stage-backoff retransmission algorithm. The density function of the time required for this new protocol to gather one packet from each sensor in a cluster is derived in this section. Numerical and simulation results for this SOSBRA-based MAC protocol are presented in Section 4. These results also show that the new protocol performs significantly better than standard 802.11 DCF and 802.15.4. Analytical results on the optimal contention window size for the new retransmission algorithm are provided in Section 5. Section 6 discusses the relative merits of TDMA and SOSBRA-based 802.11.

2. RELATED WORK

Clustered architectures for sensor networks have been studied extensively [1-11]. However, work has only recently begun on the design of Medium Access Control (MAC) protocols for these architectures. The most prominent example is an option in the proposed ZigBee standard, 802.15.4. In this option, a synchronization beam from a PAN-Coordinator (clusterhead) initiates a superframe on the wireless channel. Nodes randomly schedule their first transmissions in the 16 slots in this superframe. Subsequent new transmissions and retransmissions follow slotted-CSMA/CA procedures.

A synchronization beam and random scheduling of the first transmission attempt are also used in the SOSBRA-based MAC protocol defined in this paper. The difference is the way backoffs are handled. Specifically, failed transmissions in ZigBee can be rescheduled for either the current or subsequent superframes. In SOSBRA-based protocols, failed transmission attempts in the current “backoff window” must wait until the next backoff window to attempt a retransmission.

In [14], a scheduling algorithm for data collection is proposed. The clusterhead uses information about interference patterns created by the physical location of each non-CH node within the cluster to schedule simultaneous, collision-free transmissions. While [14] determined a lower-bound on the performance of other MAC protocols, its implementation depends on knowledge of the physical location of each node. This may not be possible in many cases and relies on either polling by the clusterhead or well-synchronized clocks in the multiple-hop nodes.

Standard 802.11 DCF has been considered for sensor nets because of its low-cost and wide availability. It does not require synchronization as in [14] and ZigBee [12] - but the cost is wasted energy due to collisions and delays due to idle channel time. More generally, 802.11 DCF was designed for homogeneous wireless peer-to-peer communication amongst large numbers of nodes, so it is far from optimal for the clustered structure that is often present in sensor networks.

The retransmission algorithm that is most closely related to the SOSBRA approach developed in this paper is the MACAW protocol [15]. MACAW assumes that each user in the network can hear every other user and that packets’ headers contain the backoff window counter setting that was used when that packet was successfully transmitted. All users set their backoff window counters to this value, and thus use a common backoff window counter after each successful transmission. A similar capability is achieved under SOSBRA by having the clusterhead transmit the backoff window size to be used by all nodes when it transmits the synchronization beam – and, unlike MACAW, we only require that all nodes be able to hear the clusterhead. Furthermore, in SOSBRA the same contention window size is used by all users until it is changed by the clusterhead, thus ensuring fairness at all times, not just after a successful transmission. Another difference is that SOSBRA-based protocols in clustered environments use an optimal backoff window size determined by the number of active nodes in the cluster, as discussed in Section 5.

Recent work on retransmission algorithms includes the Fast Collision Resolution (FCR) and Fairly Scheduled FCR (FS-FCR) algorithms proposed in [16]. In these algorithms, feedback from the channel on the number of idle slots, collisions, and successful transmissions is used by individual nodes to adjust a “multiplicative-increase, linear-decrease” backoff algorithm. At any point in time, different users may thus be using different size backoff windows. A modified collision resolution algorithm called Gentle DCF (GDCF) is proposed to enhance the performance of IEEE 802.11 DCF [17]. The contention window size is halved after c consecutive successful transmissions and an optimal value for c is proposed.

Other work on retransmission strategies [22] considers an event driven system in which N nodes that have detected an event, such as a fire, are all trying to an alert packet to a base station. The optimal probability mass function that each node should use when transmitting a packet in one of K slots is determined. The density is optimal in the sense that it minimizes the time until the first successful transmission. In our case, we fix the number of nodes, and assume transmission attempts are uniformly distributed over a window of size W. We find the value of W that minimizes the overall time to collect ALL the data packets.

3. PERFORMANCE OF STANDARD AND SOSBRA-BASED 802.11 PROTOCOLS IN A CLUSTER

In a cluster of sensors used for target tracking, nodes may only be active after a target is detected. The traffic pattern that occurs after this event is not a Poisson arrival process; instead, each of the N nodes in the 1-hop sensor cluster has data to transmit and they begin at roughly the same time. We also assume that each sensor will transmit one packet to the clusterhead and then remain silent until all other nodes have transmitted. A similar traffic pattern can be expected in environmental and building monitoring networks where each node in a cluster regularly updates the CH with one data sample. Our goal is to find a MAC protocol that is well-suited to these situations in clustered sensor networks.

MAC protocols can be sorted into two categories: distributed random access protocols, such as CSMA/CA, and centralized scheduling algorithms, such as polling and TDMA. In the class of distributed random access protocols, 802.11 DCF is currently the dominant protocol in terms of presence in the commercial market. It is thus a baseline against which alternatives should be compared. Thus, in the following subsection, we determine the time to gather one packet from each sensor in a cluster using 802.11 DCF. These results, plus the performance of a TDMA protocol, will be used later in comparisons with the new SOSBRA-based 802.11 protocol.

3.1. Standard 802.11 DCF in a Clustered Network

Previous results have shown that the four-way, RTS/CTS-based mechanism in standard 802.11 DCF is very effective in improving channel throughput, especially when large data packets are considered. In this paper, we will only consider this four-way handshaking mechanism. All simulations of 802.11 DCF in this paper are conducted with NS-2 [18]. The standard 802.11 DCF model is used in the simulations and the relevant parameters in the DCF protocol are set as in Table 1.

Table 1: Configuration of Standard 802.11 DCF for NS-2 Simulations used in this paper.

CW_{Min}	31
CW_{Max}	1023
SIFS	10 µs
Slottime	10 µs
DataRate	1M bps
RTS	352 bits
CTS, ACK	304 bits
Data Packet Length	1000 bits

Before presenting our analysis, we define all variables used in the following sections:

W : The size of the contention window; measured in slots.

$f_{N,W}$: The cost function when a contention window of W slots is used and there are N nodes.

P_S : The probability that all nodes can be successfully transmitted without any collisions.

$P_{W,n,r,c}$: The probability that n nodes transmit successfully and the other r nodes are involved in c collisions. The contention window size is W.

T_C : The duration of an RTS collision; defined as $T_C = T_{RTS} + T_{EIFS} + T_{DIFS}$.

T_D : The duration of a successful data packet transmission; defined as $T_D = T_{RTS} + T_{CTS} + T_{DATA} + 3 \cdot T_{SIFS} + T_{DIFS}$.

$T_E(N,W)$: The total time required to collect one packet from each node in the one-hop cluster when there are N nodes and the backoff window is W. For simplicity, T_E may be used.

T_W : The total time wasted on collisions and idle slots w emptying a cluster; thus, $T_W = T_E(N,W) - N \cdot T_D$.

T_i : The duration of the i -th round during the collection of one packet from each node.

T_{EIFS} : The duration of an Extended InterFrame Space; defined as $T_{EIFS} = T_{SIFS} + T_{DIFS} + T_{ACK}$.

If we assume that there is no interference from outside of the cluster and that all nodes begin their transmissions on the cue of a single event, then every node will wait to sense the channel idle for one DCF InterFrame Space (DIFS) period and transmit its RTS. The common destination node, the clusterhead (CH), will then see the collision of all of these RTS packets. Each collision results in an Extended InterFrame Space (EIFS). At this point, each node will generate a discrete random backoff slot number, which is an integer that is uniformly distributed in the interval $[0, W-1]$.

Define $\bar{W} = (W_1^1, W_2^1 ....W_{N-1}^1, W_N^1)$ to be the backoff vector the N users have chosen; each lies in $[0,\ W-1]$. Without loss of generality, we assume that $W_1^1 \le W_2^1 \le .... \le W_{N-1}^1 \le W_N^1$. Depending on the relationships among the W_i^1s, there are two cases:

1. *No collisions*. If all W_is are distinct, then all data packets can be successfully transmitted. This case occurs with probability $P_S = \frac{W \cdot (W-1) \cdot .... \cdot (W-N+1)}{W^N} = \frac{(W)_N}{W^N}$, where we assume that $N < W$, and by convention, $(W)_N = W \cdot (W-1) \cdot ... \cdot (W-N+1)$. If $N > W$, at least one collision must occur. In this case of no collisions, the total time to empty the cluster will be $T_E = T_c + W_N^1 + N \cdot T_D$. The mean value of T_E in this case can easily be shown to be:
$$E(T_E \mid \text{no collisions})$$
$$= T_c + E(W_N^1 \mid \text{no collisions}) + N \cdot T_D \ ,$$
where, recalling that there is a slot numbered zero,
$$E(W_N^1 \mid \text{ no collisions})$$
$$= \sum_{x=N-1}^{W-1} x \cdot P(W_N^1 = x \mid \text{ no collisions})$$
$$= \sum_{x=N-1}^{W-1} x \cdot \frac{N \cdot \binom{x}{N-1} \cdot (N-1)!}{W^N}.$$
2. *Collisions*. With probability $1 - P_S$, there will be at least one collision. In this case, some of the $W_i^1 s$ assumed

the same value. When their backoff slot counters reach zero, simultaneous transmissions of RTS's from all nodes with this W_i^1 will lead to a collision, which causes all busy nodes to freeze their backoff counters for one EIFS period. The nodes that collided will generate new random backoff numbers uniformly distributed in $[0, 2W-1]$.

Without loss of generality, we assume that only one collision occurs, that two nodes were involved, and that these two nodes' first stage backoff window is $W_i^1 = W_{i+1}^1$, $1 \le i \le N$. We further assume their second-stage backoff window counters are W_i^2 and W_{i+1}^2, with both in the interval $[0, 2W-1]$.

Since W_i^2 and W_{i+1}^2 are uniformly distributed over $[0, 2W-1]$ and the first stage backoff slots for the rest of the busy nodes are uniformly distributed over $[0, W-1]$, the possibility exists, especially when the number of nodes is fairly large, that W_i^2 or W_{i+1}^2 have the same value as one of the remaining backoff counters of the rest of the busy nodes in the first backoff stage. This can cause collisions that we call "cross-stage" collisions.

By the definition of the EIFS, collisions lead to long idle times on the channel, which decreases the channel throughput significantly. While increasing the window size can alleviate the problem, cross-stage collisions can not be prevented completely in 802.11 DCF.

3.2 A SOSBRA-based Approach to 802.11

The above investigations of the behavior of standard 802.11 DCF in a clustered sensor network lead us to propose an 802.11 protocol in which binary exponential backoff is replaced by a Synchronized, One-Stage-Backoff Retransmission Algorithm (SOSBRA).

3.2.1 The New Algorithm

Due to the simple clustered architecture of a sensor network, synchronization between nodes within a cluster is possible and is assumed throughout the remainder of this paper. The synchronization can be achieved via communications between the clusterhead (CH) and non-CH nodes during configuration of the cluster and by periodic synchronization updates from the CH.

With knowledge of the number of non-CH nodes in the cluster, the CH will broadcast a control packet which contains a fixed backoff window size, referred to as W in what follows, that is used by all nodes in the cluster. We assume that all the nodes of the cluster correctly receive packets with synchronization and backoff-window information. Our goal is to determine the optimal size of the backoff-window.

The transmission of data packets from every sensor node to the CH begins at time $t = 0$ with each node selecting a random number W_i that is uniformly distributed over $[0, W-1]$. We assume each node will maintain two counters. One is a backoff counter with starting value W_i; the other counter starts with value W and is called the stage counter. All nodes will start sensing the channel at time $t = 0$, decreasing their counters, both backoff and stage counters, after each slot period and freezing their counters when a transmission is detected on the channel.

When a node's backoff counter reaches zero, the node will transmit its RTS packet to the CH. If no other node has chosen the same backoff window slot, the node will successfully transmit its packet following the RTS-CTS-DATA-ACK procedure of 802.11 DCF . The remaining nodes will activate their backoff counters after one DIFS period after the packet transmission ends.

Collisions will occur if two or more nodes choose the same random backoff slot. Contrary to the standard 802.11 DCF, though, the nodes involved in the collision will *not* generate a new random backoff window until their stage counters reach zero. This ensures that all scheduled attempts to transmit RTS packets take place before any are rescheduled because of collisions.

For all the nodes within the cluster, the collision will result in an EIFS idle time on the channel while the counter of each node is frozen.

With synchronization of all the nodes in the cluster, the stage counter of every node will reach zero simultaneously. At this point, the nodes that suffered collisions will generate a new random backoff slot that is uniformly distributed over $[0, W-1]$ and begin sensing the channel.

The idea of this retransmission algorithm is that synchronization will make all nodes that were in collisions generate the next random backoff slot simultaneously and at the end of the first backoff stage, thus preventing "cross-stage" collisions. It is obvious that the channel may stay idle for several slots after each node's backoff counter reaches zero; however, as a collision leads to a rather long EIFS idle-time on the channel, the throughput will be improved, especially when the cluster contains a fairly large number of nodes. Furthermore, a one-stage backoff window size algorithm is much simpler than a multiple-stage backoff mechanism and should decrease the cost in energy of executing the protocol for each sensor node.

We next analyze the performance of this protocol. Only a single, one-hop cluster will be considered but the analysis can easily be extended to the multiple-hop cluster case. We assume that the total number of non-CH nodes is N and the size of the one-stage backoff window is W.

3.2.2 Performance of the Synchronized, One-Stage-Backoff Retransmission Algorithm (SOSBRA)

Define $\bar{W} = (W_1^1, W_2^1 \ldots\ldots W_{N-1}^1, W_N^1)$ to be the sorted random backoff slot vector, where each W_i is uniformly distributed over $[0, W-1]$. We determine the time to empty the whole cluster, called T_E. This is the time required for the CH to collect one data packet from every non-CH node within the cluster.

Before providing a general form for the density function of T_E, we analyze two special cases:

1. <u>No collisions</u>: all the packets are successfully transmitted in the first round. This occurs with probability

$P_S = \frac{W \cdot (W-1) \cdot .... \cdot (W-N+1)}{W^N} = \frac{(W)_N}{W^N}$. The total time T_E in this special case is given by $T_E = W + N \cdot T_D$, where T_D is the same as the case in 802.11 DCF.

2. *Collisions:* one or more collisions occur in the first round, and all the nodes that collided successfully transmit their packets in the second round. Define the successful node vector $\bar{N} = (N_1, N_2)$, where N_1 and N_2 are random variables that stand for the number of packets successfully transmitted in the first and second round, respectively. Thus, $N_1 + N_2 = N$.

Assume that the number of collisions, C_1, in the first round results in N_2 nodes that must retransmit. Then the random variable C_1 takes values in the set $\{1, 2, ..., \left\lfloor \frac{N_2}{2} \right\rfloor\}$. In this case, the total time to empty the cluster is:

$$\begin{aligned} T_E &= T_1 + T_2 = W + N_1 \cdot T_D + C_1 \cdot T_C + W + N_2 \cdot T_D \\ &= 2 \cdot W + C_1 \cdot T_C + \mathrm{N} \cdot T_D . \end{aligned} \quad (1)$$

with probability:

$$P\{N_1 = n_1, \mathrm{C}_1 = c_1, \text{no collisions in the second round}\}$$
$$= \frac{\binom{N}{n_1} \cdot (W)_{n_1} \cdot \binom{W-n_1}{c_1} \cdot (c_1!) \cdot S_2(N-n_1, c_1)}{W^N} \cdot \frac{(W)n_2}{W^N}. \quad (2)$$

Here $S_2(N-n_1, c_1)$ is a special case of $S_r(n,k)$, the *r*-associated Stirling number of second kind [19].

We now analyze the general case. Assume that a total of I rounds are required to empty the cluster, and assume that the vector of the number of successfully transmitting users in each round is $\bar{N} = (N_1, N_2 .....N_I)$, where $\sum_{i=1}^{I} N_i = N$. For simplicity, we also define an associated random vector $\bar{R} = (R_1, R_2 .....R_{I-1})$, where $R_i = \sum_{j=i+1}^{I} N_j$, $i = 1, 2, ..., I-1$. Each R_i is the number of remaining nodes after the i -th round that still have to transmit their packets. We also define the collision vector $\bar{C} = (C_1, C_2 .....C_{I-1})$, where C_i, the number of slots in a round in which collisions occur, can assume the values 1, 2,…. $\left\lfloor \frac{R_i}{2} \right\rfloor$, $i = 1, 2, ..., I-1$. Note that $\bar{C}$ is a vector of length $(I-1)$ because there are no collisions in the last round.

The outcome $\bar{\mathrm{n}} = (N_1 = n_1, N_2 = n_2 .....N_I = n_I)$, $\bar{c} = (C_1 = c_1, C_2 = c_2 .....C_{I-1} = c_{I-1})$ and associated $\bar{r} = (R_1 = r_1, R_2 = r_2, ..., R_{I-1} = r_{I-1})$ for $\bar{N}$, $\bar{C}$ and $\bar{R}$ occurs with probability:

$$P\{(N_1 = n_1, N_2 = n_2, ..., N_I = n_I), (C_1 = c_1, C_2 = c_2, ..., C_{\mathrm{I-1}} = \mathrm{c}_{\mathrm{I-1}})\}$$
$$= \frac{\binom{\mathrm{N}}{n_1} \cdot (W)_{n_1} \cdot \binom{\mathrm{W} - n_1}{c_1} \cdot (c_1!) \cdot S_2(r_1, c_1)}{W^N} \cdot ... \cdot$$
$$\frac{\binom{n_{\mathrm{i}} + r_{\mathrm{i}}}{n_{\mathrm{i}}} \cdot (W)_{n_{i1}} \cdot \binom{\mathrm{W} - n_1}{c_1} \cdot (c_i!) \cdot S_2(r_{\mathrm{i}}, c_{\mathrm{i}})}{W^{n_{\mathrm{i}} + r_{\mathrm{i}}}} \cdot ... \cdot \frac{(W)n_I}{W^{n_I}}$$
$$(3)$$

This outcome will lead to a total time to empty the cluster of:

$$\begin{aligned} T_E(N,W) = \sum_{i=1}^{I} T_i &= \sum_{i=1}^{I-1} (W + c_i \cdot T_C + n_i \cdot T_D) + (W + n_I \cdot T_D) \\ &= I \cdot W + (\sum_{i=1}^{I-1} c_i) \cdot T_C + N \cdot T_D . \end{aligned} \quad (4)$$

Recall that the time wasted on the channel when emptying the cluster, T_W, is defined as $T_W = T_E(N,W) - N \cdot T_D$. From equation (4) it follows that:

$$T_W = I \cdot W + (\sum_{i=1}^{I-1} c_i) \cdot T_C .$$

Note that $\{\bar{N} = (n_1, n_2, ..., n_I), \bar{C} = (c_1, \mathrm{c}_2, ..., \mathrm{c}_{\mathrm{I-1}})\}$ implies $\{T_E = I \cdot W + (\sum c_i) \cdot T_C + N \cdot T_D\}$. The converse is not true because different successful node vectors and collision vectors may lead to the same total time T_E if the sums of their collision vectors are identical. However, the probability density function of T_E can be numerically evaluated by listing all possible vector pairs $\{\bar{N}, \bar{C}\}$ that can produce a given T_E.

4. NUMERICAL AND SIMULATION RESULTS

We now present numerical and simulation results regarding the performance of the SOSBRA-based 802.11 protocol. In all cases, the time scale is normalized by a slot time, which is equal to 10µs (see Table 1). Note that the time of successful transmissions on the channel is common to all CSMA-based protocol, that is $N \cdot T_D$. Minimizing the wasted time T_W will thus minimize delay and maximize channel utilization.

In Fig. 1 we plot the probability mass function of T_E for $N = 50$ and $W = 120$. Note that T_E is significant over a broad range: the interval [12000, 15000].

In Fig. 2, simulation results for the average channel throughput ρ, defined as $N \cdot T_D / T_E = (N \cdot T_D)/(N \cdot T_D + T_W)$, with different backoff window sizes are provided. All results are averaged over 100,000 runs. From the figure, it is clear that ρ achieves a maximum value at some backoff window size W_{opt}. As we mentioned earlier, $T_E = T_W + N \cdot T_D$ and the term $N \cdot T_D$ is common to all the protocols. From equations (1) and (4), T_W consists of two terms: the time due to collisions and the time due to empty slots. While increasing the backoff interval may increase the probability of successful transmission of data packets in each mini slot, it also increases the time until the end of the round. On the other hand, decreasing the backoff interval will decrease the time spent waiting to the end of the round while increasing the wasted time due to collisions. The trade off between these two effects leads to an optimal backoff window size for a fixed number of sensor nodes within the cluster, as shown in Fig. 3. In Fig.2, It shows that overall channel throughput can reach as high as 0.92 for all four cases in the figure, and this proves the high time-efficiency of SOSBRA-based protocol. We further note that the optimal throughput is not very sensitive to the contention window size in a wide range, especially for the case of larger number of nodes. For example, for the case of $N = 100$, when $W = 600$, that is one and half times of optimal window size, the overall channel throughput does not change much. This offers great flexibility for system design.

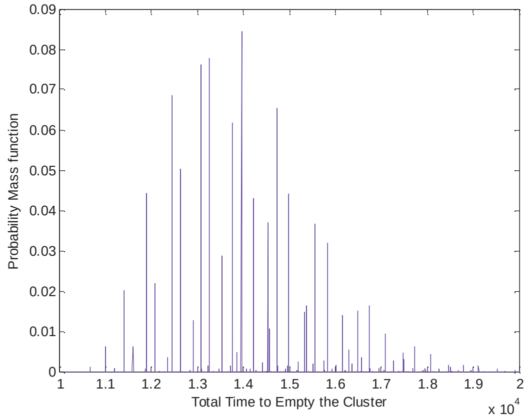


Fig. 1: *Numerical results for the probability mass function of* T_E*, the total time to empty the cluster, for the SOSBRA-based 802.11 protocol. Here,* N *=50 nodes and* W *=120.*

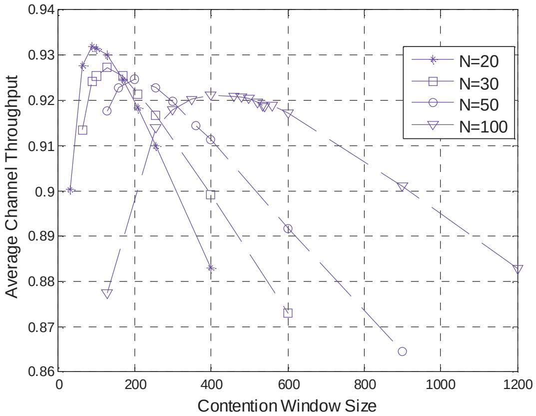


Fig. 2: *Simulations for the SOSBRA-based 802.11 protocol that show the average channel throughput during the emptying the cluster for different contention window sizes.* N *is the number of nodes in the cluster.*

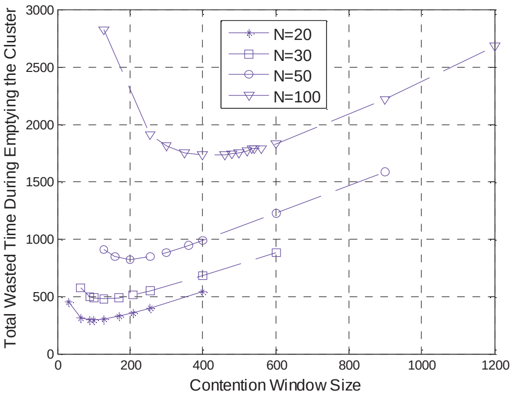


Fig. 3: *Simulations for the SOSBRA-based 802.11 protocol that show the average wasted time,* T_W*, during empty the cluster for different contention window sizes.* N *is the number of nodes in the cluster.*

It is interesting to investigate the relationship between the optimal backoff window size W_{opt} and the total number of sensor nodes within the cluster. In Fig. 4, we plot the optimal backoff window sizes for different cluster sizes. The corresponding results for the total time to empty the cluster are shown in Fig. 5. Note the nearly linear relationship between the optimal backoff window size and the total number of nodes. It provides a simple way to estimate the optimal window size when the CH knows the total number of nodes within the cluster.

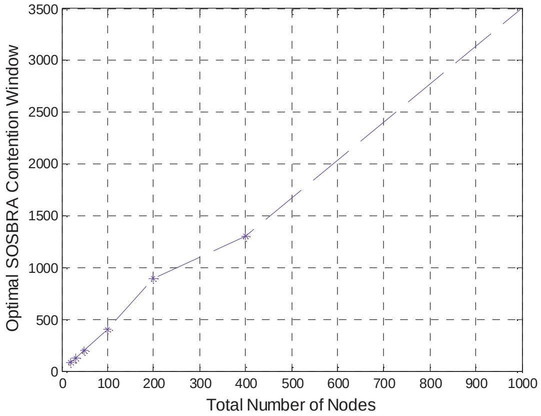


Fig. 4: *Simulations determining the optimal contention window size for different* N *for the SOSBRA-based 802.11 protocol.*

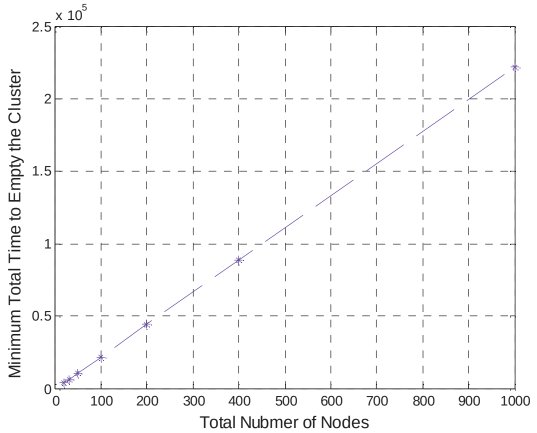
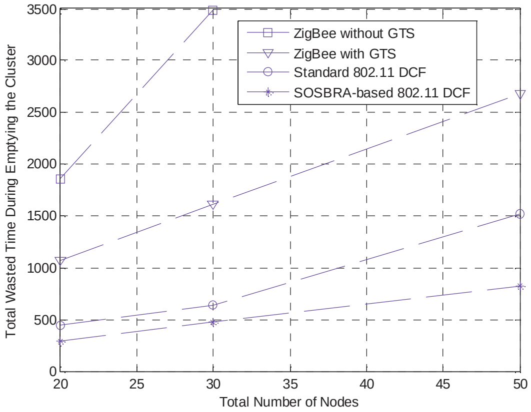


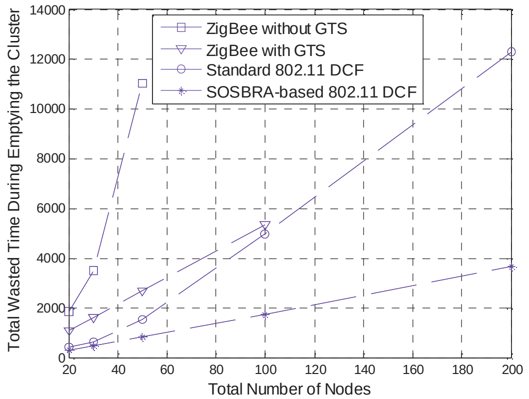
Fig. 5: *Simulations determining the minimum T_E, the minimum total time to empty the cluster, for different cluster sizes for the SOSBRA-based 802.11 protocol.*

Fig. 6 provides comparisons between the SOSBRA-based 802.11 protocol, standard 802.11 DCF, and ZigBee with and without GTS. For fair comparisons, we record wasted time T_W only since ZigBee without GTS defines a different ACK(acknowledgement) packet size from that in 802.11 DCF. In each simulation of ZigBee a superframe is used. Different superframe orders are tested and the superframe order that yields the best performance – lowest channel wasted time – is chosen. For ZigBee with Guaranteed Time Slots (GTS), we ignore the overhead due to GTS allocations and reallocations, so the actual performance of the scheme will be worse than the results presented here.

From the figure, it is clear that the SOSBRA-based 802.11 protocol utilizes the channel better than both 802.11 DCF and ZigBee. Its advantage increases as the number of nodes increases.



(a). Case of small number of nodes



(b). Case with large number of users

Fig. 6: *Simulations comparing the wasted-time T_W before the cluster is emptied for the SOSBRA-based 802.11, Standard 802.11 DCF, and ZigBee with and without GTS. The advantage of the SOSBRA-based approach is significant for all cluster sizes but increases as the number of nodes in the cluster increases.*

The reason for this improved performance is the elimination in the SOSBRA approach of "cross-stage" collisions, a phenomenon that becomes dominant in 802.11DCF and ZigBee without GTS as the number of nodes increases. Though ZigBee with GTS may prevent "cross-stage" collisions, some portion of the superframe must be dedicated to Contention Access Period (CAP); furthermore, each node must reserve an integer number of time slots for data packet transmission. These requirements create significant overhead, which shows up in the figure as inefficient use of the channel.

Energy-efficiency is also a critical issue in sensor networks. Note that the performance improvement of SOSBRA-based 802.11 DCF over standard 802.11 DCF and ZigBee comes from the elimination of "cross-stage" collisions. Each collision involves at least two transmissions, and nodes consume energy even while waiting until they can retransmit, so we can conclude that SOSBRA is more energy-efficient than 802.11 DCF and ZigBee. To make these comparisons preceise, we use some published measurements on energy consumption ratios when nodes are idling, receiving and sending: The Digitan 2Mbps Wireless module (IEEE 802.11/2Mbps) specifications shows idle:receive:send ratios of 1:2:2.5 [20]. We apply these ratios and calculate the total energy consumed to empty a cluster with different protocols. The results are shown in Fig. 7.

In Fig. 7, all the energy consumed is normalized by energy consumed by a node when it is idle for 1000 microseconds. From Fig. 7, we can see that SOSBRA-based 802.11 DCF is the most efficient in term of energy consumption. In Fig. 6, we can see that standard 802.11 DCF is more time-efficient than ZigBee with GTS. However, in terms of energy efficiency, ZigBee with GTS outperforms standard 802.11 DCF. This is because there are no collisions in ZigBee with GTS, and a node being idle consumes less energy than one that is sending during the same time period.

Furthermore, each collision involves at least two sending events.

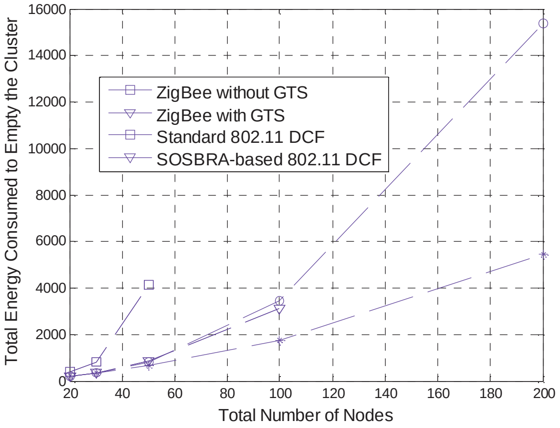


Fig. 7: *Simulations comparing total energy consumption to empty the cluster for the SOSBRA-based 802.11, Standard 802.11 DCF, and ZigBee with and without GTS. The energy consumption ratios used in these simulations was idle:receive:send=1:2:2.5*

5. OPTIMAL CONTENTION WINDOW SIZE FOR SOSBRA-BASED PROTOCOLS

From the sections above, it is obvious that an appropriate choice of the contention window size W, is critical to the SOSBRA-based 802.11 protocol's performance. In the extreme case of $W = 1$, all nodes will collide with each other and no packets would ever reach the clusterhead. On the other hand, if W is so large that almost all packets can be successfully transmitted to the clusterhead on their first attempt, the idle time between successive packet transmissions would be very large. Thus, choosing W involves a trade-off between the wasted time caused by collisions and idle slots left by long contention windows.

5.1. The Optimal Contention Window Size

Given the total number of nodes, N, and the window size W, the mean wasted time to empty the whole cluster, $E\{T_E(N,W)\}$ is used to evaluate the performance of the algorithm:

$$E\{T_E(N,W)\} = \sum_{n_1=0, n_1 \neq N-1}^{N} \sum_{c_1=1}^{\left\lfloor \frac{N-n_1}{2} \right\rfloor} \mathrm{T}_{W,\, n_1,\, N-n_1,\, c_1} \cdot P_{W,\, n_1,\, N-n_1,\, c_1}$$

where $T_{W,n_1,N-n_1,c_1} = W + c_1 \cdot T_c + E\{T_E(N-n_1, W)\}$

(5)

Determining the optimal contention window size through equation (5) is difficult, so we present another approach to estimate the optimal contention window size. As numerical and simulation results show later, this approach provides a reasonable estimate of the optimal window size. The nature of the approach is to minimize a cost function that is defined below.

For any one of the W slots in a backoff interval, three possible cases may happen:

1) No nodes choose this slot; this occurs with probability $P_{empty} = (1-\frac{1}{W})^N$.
2) Only one node chooses this slot; this occurs with probability $P_{succ} = N \cdot \frac{1}{W} \cdot (1-\frac{1}{W})^{N-1}$.
3) More than one node choose this slot; this occurs with probability $P_{coll} = 1-(P_{empty} + P_{succ})$.

Define $f_{N,W}$ to be the cost function when contention window W is used, which characterizes the wasted time during collecting data packets. This cost comes from two sources. The first is from the number of idle slots in the window of size W; the other one comes from possible collisions. We thus define

$$f_{N,W} = W + P_{coll} \cdot (T_C + \frac{f_{N,W}}{W}) \cdot W. \qquad (6)$$

Equation (6) is derived as follows. Given N and W, the average length for any of the W slots in the first round is $1 + P_{coll} \cdot T_c$. The total average length of the first round is $W \cdot (1 + P_{coll} \cdot T_c)$ given that that the W slots are independent from each other. Furthermore, users will enter subsequent rounds with probability P_{coll}. If we overestimate the number of collided users by assuming there are N of them, which is the worst case, equation (6) follows.

The term $\frac{f_{N,W}}{W}$ arises because one collision in a slot will cause the colliding nodes to reschedule transmissions for the second round, inducing additional costs. Since $f_{N,W}$ is the total cost for all W slots, on average, the "collision cost" for each slot is $\frac{f_{N,W}}{W}$.

It is worthwhile noting that (6) overestimates the effect of collisions because we did not consider the cost of successful transmissions. Actually, one collision will contribute less than $\frac{f_{N,W}}{W}$ to the cost in the following round because there will almost always be fewer than N nodes in that round. Thus, the contention window size we get through this method should be larger than the optimal one. However, to simplify our analysis, we ignore that effect.

From (6), we obtain

$$f_{N,W} = \frac{W + T_C \cdot W \cdot (1-(1-\frac{1}{W})^N - N \cdot \frac{1}{W} \cdot (1-\frac{1}{W})^{N-1})}{(1-\frac{1}{W})^N + N \cdot \frac{1}{W} \cdot (1-\frac{1}{W})^{N-1}}.$$

(7)

Taking the derivative with respect to $V = \frac{1}{W}$,

$$f_{N,V}{}' = \frac{T_C}{V^2} - \frac{(T_C+1) \cdot ((-N^2+1) \cdot V^2 + (N-2) \cdot V + 1) \cdot (1-V)^{N-2}}{V^2 \cdot (1-V)^{2N-2} \cdot (1+(N-1) \cdot V)^2}.$$

(8)

It is not straightforward to determine the V_{opt} such that $f'_{N,W} = 0$ from equation (8). We plot $f_{N,W}$ vs V over $(0,1/N)$ in Figure 8.

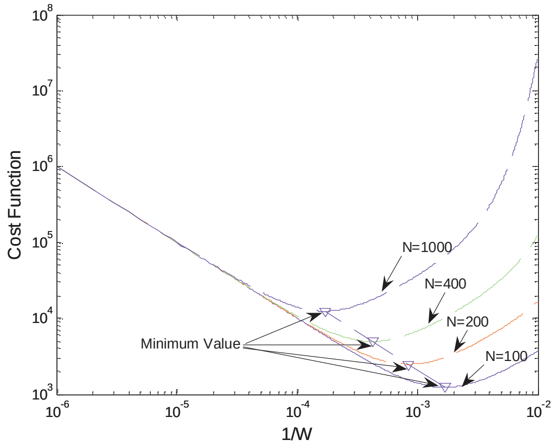


Fig. 8: *Numerical results showing the cost function* $f_{N,W}$ *vs* $1/W$ *for different cluster sizes. The cost function assumes a minimum value in* $[0,1/N]$ *for different* N.

In Fig. 9 we plot the optimal contention window size obtained from simulations and the above analysis. As we stated above, the analytical results yield a slightly larger optimal contention window size than the simulation results.

In Fig. 10, we compare the mean time to empty the cluster with the optimal contention window size from the simulation and analytical results. From the figure, we found that the mean total times agree with each other well, even though the analytically chosen backoff window size is not optimal. This follows from the fact that the curves of the mean total time to empty the cluster vs. the contention window size are quite flat around their minimal points. Thus, even though the analysis gives a slightly larger contention window, this window still achieves nearly optimal performance.

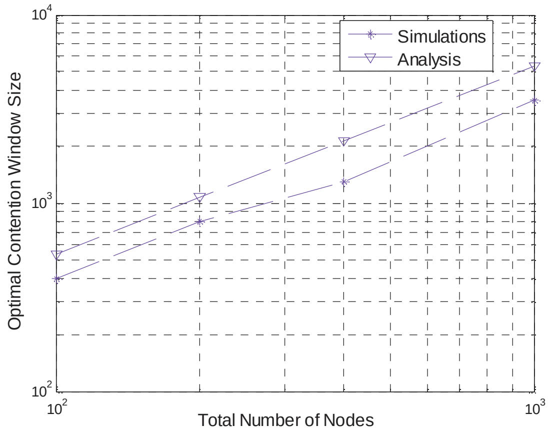


Fig. 9: *Comparison between simulation and analytical results for* W_{opt} .

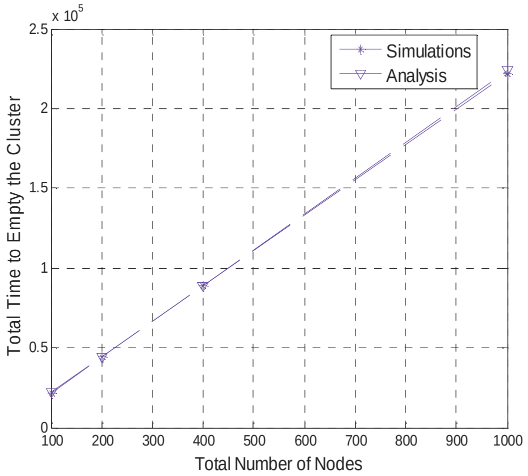


Fig. 10: *Average Total time obtained with* W_{opt} *from both simulation and analysis.*

5.2 Optimal Contention Window when the Number of Nodes is Large

In sensor networks, sensor nodes are usually deployed with high density to ensure adequate spatial sampling resolution and redundancy. In this subsection, we investigate the optimal contention window for the case of a very large number of nodes within the cluster. We maintain that the optimal contention window size in a SOSBRA-based approach is a linear function of the total number of nodes.

As is well known [21], binary exponential backoff retransmission algorithms are ultimately unstable for large user populations in the sense that the asymptotic rate of successful transmission is zero, no matter how small the arrival rate. In other words, for a system with binary exponential backoff, the ultimate queuing time for data packets must eventually increase to infinity. In our proposed SOSBRA approach, as the number of nodes increases, the contention window size must increase correspondingly. From Section 3.1, it is known that if W is a linear function of N, say $W = \alpha \cdot N$, the probability of a collision in any slot converges to a constant:

$$\begin{aligned} &\lim_{N\to\infty, W=\alpha\cdot N} P_{coll} \\ &= \lim_{N\to\infty, W=\alpha\cdot N} (1-(1-\frac{1}{W})^N - N\cdot\frac{1}{W}\cdot(1-\frac{1}{W})^{N-1}) \\ &= 1-e^{-\frac{1}{\alpha}} - \frac{1}{\alpha}e^{-\frac{1}{\alpha}}. \end{aligned}$$

Recall that our measure of cost was defined to be:

$$f_{N,W} = \frac{W + T_C\cdot W\cdot(1-(1-\frac{1}{W})^N - N\cdot\frac{1}{W}\cdot(1-\frac{1}{W})^{N-1})}{(1-\frac{1}{W})^N + N\cdot\frac{1}{W}\cdot(1-\frac{1}{W})^{N-1}}$$

Now, define $f^1_{N,W} = \frac{f_{N,W}}{W}$ to be the cost per slot. If we set $W = \alpha\cdot N$ and let N converge to infinity, we obtain

$$\begin{aligned}&\lim_{N\to\infty,W=\alpha\cdot N} f^1_{N,W}\\&=\lim_{N\to\infty,W=\alpha\cdot N}\left[\frac{1+T_c}{(1-\frac{1}{W})^N+\frac{N}{W}(1-\frac{1}{W})^{N-1}}\right]-T_c\\&=\frac{1+T_c}{e^{-\frac{1}{\alpha}}+\frac{1}{\alpha}e^{-\frac{1}{\alpha}}}-T_c=C(\alpha)\ .\end{aligned}\tag{9}$$

From (9), it is clear that, for very large N, we may approximate the total cost $f_{N,W}$ as:

$$f_{N,W}=W\cdot f^1_{N,W}=N\cdot\alpha\cdot C(\alpha)\ .\tag{10}$$

This further leads to the overall channel throughput to empty the cluster as:

$$\rho=T_W\ /T_E=\frac{f_{N,W}}{f_{N,W}+N\cdot T_D}=\frac{\alpha\cdot C(\alpha)}{\alpha\cdot C(\alpha)+T_D}\tag{11}$$

Equations (9) and (10) show that the cost per slot converges to a constant that is a function of α only. The total cost is then a linear function of N. And this results in a constant channel throughput. The α that minimizes $f_{N,W}$ is independent of N, and can be easily evaluated by taking derivatives of $\alpha\cdot C(\alpha)$.

Fig. 11 shows the relation between T_C and the α that minimizes $f_{N,W}$. It is clear that α increases as T_C increases. As explained above, T_W comes from two sources: idle backoff slots and time from possible collisions. When T_C increases, a possible collision has a more severe effect on T_W. A larger contention window in this case will reduce the average number of collisions in each round; thus minimizing the effect of collisions on the overall T_W.

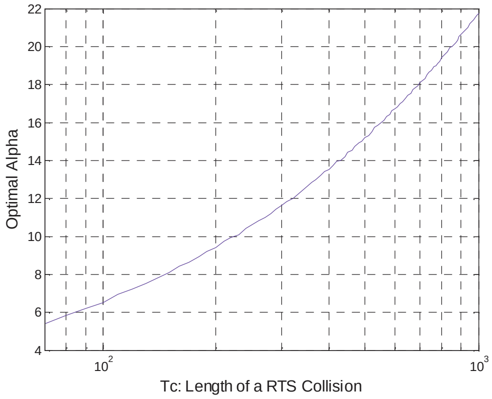


Fig. 11: *Relationship between Optimal* α *and* T_C.

6. SOSBRA AND TDMA: A COMPARISON

The SOSBRA-based 802.11 protocol introduced above and ZigBee both rely on synchronization of the motes in a cluster. When synchronization is possible, centralized protocols such as TDMA and polling are also possible and should be evaluated. We quickly eliminated polling because it would require so many transmissions from the clusterhead that it would quickly exhaust its battery.

We then considered TDMA and evaluated its performance according to the following criteria:

1) Complexity: Each node is assigned a distinct slot in TDMA. This requires the clusterhead to know the identity of each mote in the cluster in order to order them. It must broadcast the ordering to the cluster in order to assign slots and should likely repeat this broadcast each time a mote dies from lack of energy or the network is re-clustered.

2) Inflexibility: Each node in TDMA can only transmit one fixed-length packet per frame. If the packet is shorter than a slot, the remaining time in the slot is wasted. If a node has a packet longer than a slot, then the packet must be split and transmitted over two frames, causing excessive delay. Random access schemes are much more flexible in accommodating variable packet lengths. Variable packet lengths are likely because of compression algorithms used to reduce the number of bits, and hence the energy, to transmit a packet.

3) Fault Tolerance: Sensor nodes are usually cheap and vulnerable to failure; furthermore, they will all eventually stop working because they have depleted their batteries. It is difficult for the CH to detect such unpredictable failure events. In this case, pre-assigned time slots for failed nodes become wasted which is costly because each slot in TDMA is as long as a data packet. 802.11, both with and without SOSBRA, is not very sensitive to the change of node number in the cluster, as shown in Figure 2. In Figure 12, we provide the performance comparison between SOSBRA and TDMA-based approach when the cluster topology changes due to either node failures or other changes in topology. p stands for the probability that a node has failed. Here we assume the total number of nodes in the cluster is $N=1000$, the data packet length is $PL=1000\,bits$, and all other parameters are same as specified in Table 1. In SOSBRA, we fix the contention window size $W_{opt}(N)$ for $N=1000$ for all of cases. The figure shows that when the node failure probability is larger than 0.15, TDMA performs worse than SOSBRA. Furthermore, we ignore the overhead for broadcasting slot assigning information in TDMA, so the TDMA approach actually performs even worse than that shown here. p can also be interpreted as the probability that a node does not has a packet to send during the sampling process. We conclude that SOSBRA is more fault-tolerant compared to TDMA-based approach.

SOSBRA is not very sensitive to changes in the contention window size, especially when the number of nodes is large. In other words, for a fixed contention window size, changes of the number of nodes participating in the sampling process do not significantly affect the overall performance. Thus, even if nodes are added or some skip the initial rounds of data collection, the overall performance – in terms of the overall time to empty the cluster – does not change much. Thus SOSBRA offers much flexibility than TDMA with regard to changes in topology or traffic patterns.

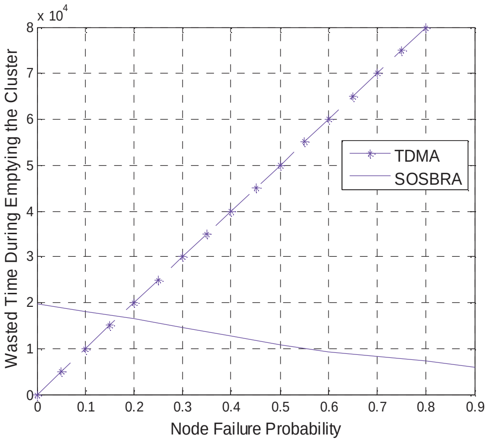


Fig. 12: *Comparison between SOSBRA and TDMA-based approaches. Here* $N = 1000$, $PL = 1000\,bits$ *and a slot time is 10 microsecond in SOSBRA.*

7. CONCLUSIONS

In this paper, we proposed a synchronized, one-stage-backoff, retransmission algorithm (SOSBRA) for MAC protocols for clustered sensor networks. We studied the performance of this new algorithm in 802.11 via both probabilistic analysis and simulation. We showed via simulation that a SOSBRA-based 802.11 DCF protocol performs significantly better - in terms of energy usage, delay, and throughput - than both 802.15.4 (ZigBee) and 802.11 DCF (WiFi). It is superior to TDMA in terms of flexibility and complexity; especially from the point of view of the mote serving as the clusterhead.

8. ACKNOWLEDGEMENT

This work was supported in part by the Indiana 21'st Century Research and Technology Fund.

9. REFERENCES

[1] S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proceedings of IEEE INFOCOM'03*, San Francisco, April 2003.

[2] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *Journal on Selected Areas in Communication*, Vol. 15 pp.1265-1275, September 1997.

[3] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310-315, June 1999.

[4] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," *Proceedings of the Vehicular Technology Conference*, Vol. 2, pp. 889-893, 1999.

[5] M. Gerla, and J. T. C. Tsai, "Multicluster, Mobile, Multimedia Radio Networks," *Wireless Networks*, Vol. 1, No. 3, pp. 255-265, 1995.

[6] A. B. McDonald, and T. Znati, "A Mobility Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1466-1487, Aug. 1999.

[7] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, Vol. 29, No. 11, pp. 1694-1701, Nov. 1981.

[8] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks," *Journal of Cluster Computing*, Special issue on Mobile Ad hoc Networking, No. 5, 2002, pp. 193-204.

[9] S. J. Baek, G. D. Veciana and X. Su, " Minimizing Energy Consumption In Large-scale Sensor Networks Through Distributed Data Compression And Hierachical Aggregation," *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 6, pp. 1130-1140, Aug. 2004.

[10] O.Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," *Proceedings of IEEE INFOCOM'04*, Hong Kong, March 2004.

[11] V. Mhatre and C. Rosenberg; "Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation," *Ad Hoc Networks Journal*, Elsevier Science, Volume 2, Issue 1, Pages 45-63, Jan. 2004

[12] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems –Local and metropolitan area networks –Specific Requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS)," IEEE Standard 802.15.4, Oct. 2003.

[13] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems – Specific Requirements – Part 11: Wireless LAN MAC and PHY Specifications," IEEE Std 802.11-1999, IEEE, New York, 1999.

[14] S. Bandyopadhyay and E. J. Coyle, "Spatio-Temporal Sampling Rates and Energy Efficiency in Wireless Sensor Networks," *Proceedings of IEEE INFOCOM'04*, Hong Kong, March 2004.

[15] V. Bharghavan, "MACAW: A Media Access Protocol for Wireless LANS," *Proceedings of SIGCOMM '94*, London, England, pp. 212-225, August 1994.

[16] Y. Kwon, Y. Fang and H. Latchmari, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," *Proceedings of IEEE INFOCOM'03*, San Francisco, April 2003.

[17] C. Wang; B. Li and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Transaction on Vehicular Technology*, Vol. 53, No. 4, pp.1235-1246, July 2004.

[18] http://www-nrg.ee.lbl.gov/ns/#version2

[19] L. Comtet, "Advanced Combinatorics," D.Reidel Publishing Company,1974.

[20] Oliver Kasten, Energy Consumption Model, Eldgenossische Technische hochschule Zurich.

[21] D. J. Aldous, "Ultimate Instability of Exponential Back-Off Protocol for Acknowledgement-Based Transmission Control of Random Access Communication Channels," *IEEE Transactions on Information Theory*, Vol. IT-33, No. 2, pp. 219-223, Mar. 1987.

[22] Y.C. Tay, K. Jamieson and H. Balakrishnan, "Collision-Minimizing CSMA and its Applications to Wireless Sensor Networks," IEEE J. Selected Areas in Communications, Vol. 22, No. 6, Aug. 2004, pp. 1048-1057.