J. Bangladesh Agril. Univ. 11(2): 321–330, 2013 ISSN 1810-3030

Wind turbine monitoring system using wireless sensor networks

M. M. Bhuiyan and M. M. Billah
Department of Computer Science and Mathematics, Bangladesh Agricultural University, Mymensingh-2202, Bangladesh, E-mail: masumuzzaman@yahoo.com

Abstract

Wireless sensor networks can be used in wind farm monitoring where periodic data collection about the sound generated inside the farm as well as detection and monitoring of faulty wind turbines is necessary. Periodic sound data collection requires reliability while faults detection necessitates timeliness. Simultaneous data gathering and faults monitoring was not well studied in literature. This paper proposed a system model that worked on homogeneous data gathering Wireless sensor networks deployed in wind farms. When a wind turbine became faulty, a cluster with a different transmission channel around that wind turbine was formed and both periodic sound data gathering and faults monitoring were performed at the same time. The proposed model had a novel routing strategy with a built-in congestion control technique to provide timely delivery of faults data. Experimental results show that the proposed method performed better than known similar techniques in terms of reliable data gathering and reliable timely faults monitoring. Due to lower number of high power transmissions, the proposed method had 8% to 17% higher success rate of regular system and 94% of accuracy at the fault monitoring. In terms of timely faults detection and notification, this method had a comparative performance to the existing methods.

Keywords: Wind turbine, Monitoring system, Wind turbines, Wireless sensor network

Introduction

Wireless sensor networks (WSNs) can be used in data gathering applications as well as event detection/ monitoring applications (Pottie, 1998; Biagioni & Sasaki, 2003; Mainwaring *et al.*, 2002; Banerjee *et al.*, 2008; Li *et al.*, 2008; Xu *et al.*, 2010; Zhu & Ni, 2008). Major research works focused on energy utilization issue (Akyildiz *et al*., 2002). However, in this experiment, proposed system deployed WSNs in wind farm monitoring applications where data collection and event monitoring persist. A wind farm covers a large geographic area having many wind turbines positioned in rows. As a typical separation distance between wind farms and residential areas is of the order of 1000 meters (Sonus Pty Ltd., 2010), the low frequency sound emitted from wind farms is dominant in dwelling areas. But this sound is audible if the sound pressure levels are above certain thresholds. Detailed values of the audible thresholds at different frequencies can be found in Moorhouse *et al.*, 2005. Infra and low frequency sound above certain pressure level may cause mental and physiological damage (Rogers *et al.*, 2006). But the pressure levels of infra and low frequency sound emitted from wind farms are below the thresholds harmful to human in residential areas (Colby *et al.*, 2009) because of atmospheric absorption of sound over a long separation distance.

Different frequencies generated from wind turbines have some sort effect on the plants grown in nearby region. A WSN can establish for periodic collection of data about pressure levels of sound of different frequencies over a long period. After statistical analysis of the data collected, it may be able to determine the adverse (or beneficial) impact on different plant species by comparing their growth status with the growth status of same species in same climatic areas away from wind farms. The same WSN can be utilized to monitor the malfunctioning of wind turbines also. If a wind turbine malfunctions then it will emit a sound different from the normal operating one. As soon as the sensor nodes detect this type of abnormal sound, they can inform the BS which will take immediate action to solve the problem.

Materials and Methods

The research considered WSNs engaged in continuous gathering of sound data as well as continuous monitoring of faulty wind turbines simultaneously. In data gathering task, delay was not crucial. But was wanted to delivery of as much data as possible for statistical analysis at further stages. On the other hand, faults notification delay was vital and should be kept as low as possible.

Nodes were manually deployed with a density of λ nodes per square meter. All nodes remained static, found own locations using localization techniques and informed the base station (BS) about their locations. During this message-exchange phase, each node determined the locations of its neighbours. Each node had two sensing devices. One was utilized in periodic sound data collections with a fixed sensing period of SP_s. Data were transmitted to the BS without considering delay. Other sensing device was used in fault sensing with a sensing period of SP_f independent of SP_s ($SP_s >> SP_f$). When a node detected a faulty wind turbine, sensing period of this device of that particular node reduced to 0.5 * SP_f. During the transmission of periodic sound data, nodes used same fixed transmission power to achieve a transmission range of d_s meters. Nodes that transmitted faults data used variable transmission power in different conditions. Each node could transmit and receive data in two different frequency bands, f_s and f_f. When a wind turbine became faulty, all surrounding nodes formed a *cluster* and one *cluster head* (*CH*) was selected. Operation inside the cluster was periodic. In every period, each node inside the cluster sent its fault decision directly to *CH* which sent the aggregated decision to the BS through multi-hop communication. Only the intra-cluster communications used frequency band f_f. All other communications (including *CH* to non-cluster nodes communication) used frequency band f_s. System used contention based MAC protocol IEEE 802.11 DCF in whole part of this protocol but with slight modification inside the cluster around the faulty wind turbine.

Cluster Formation: In *normal mode*, the fault sensing device of each node had a sensing period of SP_f. When a node's data reading crossed the lower threshold Th_1, the node reduced sensing period to $0.5SP_f$ and turned into *critical mode*. In critical mode, the node stopped data gathering operation and switches to frequency band f_f. Sensing period was reduced to lower the latency of detection of possible faults. When the upper threshold Th_u was crossed, that particular node started *cluster formation* process. Each critical node participated in cluster formation process and tried to be the *CH*. The physical property of the targeted fault was such that it could be sensed by a sensor node which was at most d_f meters away from the faulty wind turbine. Any node that took part in cluster formation process adjusted its transmission range to $2\times d_f$ so that all nodes that were sensing the faulty wind turbine formed a single collision domain. Each node, whose data reading crossed Th_u, turned its receiver on and broadcast a message claiming itself to be the *CH* (called this message *CH-claiming* message). As all such nodes constituted a single collision domain, the transmitting node's receiver received the original message if there was no collision. If there was a collision, the transmitting node did back-off process and broadcast a new *CH-claiming* message. This process was continued by all critical nodes until any node succeeded. That successful node waited *SIFS* time and then without any back-off process broadcast another message telling all nodes that it is the *CH*. This message reached collision free to all other critical nodes because they must waited at least *DIFS* time before transmitting anything again (for details of IEEE 802.11 DCF, please see IEEE Standard, 2007.)

Cluster Operation: The *CH* partitioned its operation into rounds of adaptive durations. At the beginning of each round, the *CH* broadcast a message (*round initiation* message) in frequency band f_f telling all nodes to broadcast their faults decisions. Number of nodes (n) that are monitoring the fault inside the cluster was $\pi(d_f)^2\lambda$. Nodes transmitted positive fault data only. If they could not detect the faulty wind turbine, they did not send any data to *CH*. The *CH* required positive data from 5 nodes to be sure that there is a fault. As sensor nodes' transmission was susceptible to random failure, system required that 10 nodes should broadcast positive data to *CH*. If $n > 10$, each node generated a random number δ_1 such that $0 < \delta_1 < 1$. Now if $\delta_1 > \delta$ then the node broadcast its decision. The δ was choose in such a way that $(1 - \delta)\times n \approx 10$. The value of δ depended on λ and was instructed to all nodes during network setup phase. Each critical node turned its receiver on and repeatedly broadcast its data until it was successful. A critical node, waiting to successfully broadcast data, stopped if it found 5 successful transmissions already occurred. Each node examines its last three readings. If at least 2 of last 3 readings were above th_u then that node tried to broadcast its fault decision, otherwise it remained silent. Probability of error of a sensor's reading (P_{err}) was independent of its previous readings and was quite low. Therefore, the probability of error of 2 readings within last 3 measurements was very low and can be ignored. If the *CH* received 5 positive fault decisions within T_R seconds then it sent positive fault decision towards the BS using frequency band f_s. In this case, *CH* adjusted its transmission range to $3\times d_f$ meters to surpass the cluster surrounding the faulty wind turbine. After transmitting a positive decision to the BS, the *CH* started a new round by broadcasting *round initiation* message. If the *CH* did not receive 5 positive decisions within T_R seconds in a particular round then it did not send any decision to the BS in that round and simply started a new round. It may be noted that when a node was sensing any probable fault (i.e., in critical mode), that node stopped sending periodic sound data. Actually, faults were rare and only a few nodes were involved in detection of the current fault. The BS was able to interpolate the missing periodic data of these nodes by using the data sent by these nodes before and after the fault. Moreover, as SP_s >> SP_f and the BS took initiative to diminish the fault as soon as it was notified, the amount of missing sound data was not high. Any time, when a critical node's reading fell below Th_l, that node became a normal node with sensing period of SP_f.

Routing: The proposed system regarded periodic sound data as regular data and data about faulty wind turbines as critical data. Regular data was not delay sensitive whereas critical data was highly delay constrained. When a node generated a critical data packet, it inserted a deadline time into the packet. Each transmitting node checked the deadline of the associated packet and took necessary steps to ensure that the packet would be reached the BS within its deadline. Each node's MAC layer placed incoming critical packets into its transmission queue according to non-decreasing order of their deadlines. Regular packets have no deadline field and were placed at the rear of transmission queue. A node distinguished a regular packet from a critical one by examining the existence of the deadline field. All packets (regular as well as critical) were forwarded to a single BS.

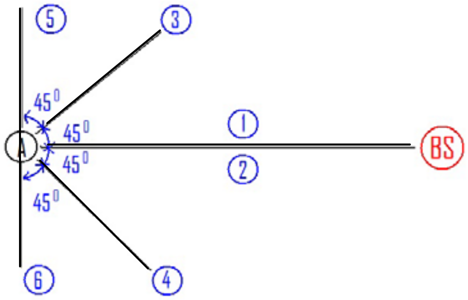

Fig. 1. Six neighbours nodal arrangement

Routing of regular packets: When a node transmitted regular data, it adjusted its transmission range to d_s meters. Each node knew locations of its neighbours. Each node *A* selected 6 neighbours (Fig. 1) to forward its data towards the BS. These nodes were data neighbours of node *A*. All data neighbours were approximately d_s meters away from node *A* and were numerically numbered from 1 through 6. Data line was defined to be the line connecting node *A* to the BS. Neighbours 1 and 2 were close to the data line. Neighbours 3 and 4 created approximately 45^0 angles with data line and are opposite to each other with respect to data line. Similarly, neighbours 5 and 6 created approximately 90^0 angles with data line. When a node found all of its data neighbours congested or unreachable, it broadcast a back pressure message (*BP-M*). After receiving this *BP-M*, all nodes stopped data forwarding to this congested node for a certain duration. After that duration they again considered this node for data forwarding. Now, consider how node *A* forwarded data towards the BS. If node *A* had already not received *BP-M* from either of nodes 1 and 2 then it transmitted each packet randomly to node 1 or 2. If either of nodes 1 and 2 broadcast *BP-M* then node *A* considered the other node. If both of nodes 1 and 2 broadcast *BP-M* or three successive transmissions by node *A* failed then node *A* forwards all data to either node 3 or node 4 (chosen randomly) provided that node had not broadcast *BP-M* yet. If node *A* chose node 3 and afterward it received *BP-M* from node 3 or its three successive transmissions failed then it chose node 6 provided that node 6 had yet to send any *BP-M*. But if node *A* chose node 4 in previous step then it would choose node 5 in similar case. The philosophy behind the jumping from node 3 to node 6 was that, when node 3 was congested there was a high chance that node 5 was also congested as they were close to each other. But node 6 was long away from node 3 and therefore, has low chance of being congested. The same was true for pair 4 and 5. When node *A* received *BP-M* from node 5 (or from node 6, which one was applicable) or its three successive transmissions failed then it broadcast *BP-M*, and all neighbouring nodes avoided forwarding their data to node *A* for a particular time duration. If node *A* was also generating data then it will reduce or stop data generation in this case.

Routing of critical packets: Each node's MAC layer calculated exponential average value of its medium access delay (*MAD*) regardless of whether the node forwarded regular or critical or both types of packets. Incoming critical packets were placed at the front part of transmission queue according to non-decreasing order of their deadlines. Before trying to send a critical packet, the MAC layer first checked how much remaining time (*RT*) was available within which this packet should reach the BS. If *RT* < *MAD* then MAC layer dropped the packet. If *MAD* < *RT* < 2.5× *MAD* and the BS was more than d_s meters away then MAC layer made node's transmission range reachable to the BS; otherwise, transmission range was adjusted to d_s meters. When the critical packet was tried using regular transmission range (d_s meters), the same routing techniques were used as those utilized in routing of regular data but with one exception. During routing of regular data, a node reduced or even stopped data transmission after it broadcast *BP-M* when it found all data neighbours (1 through 6) unreachable. Here, the node continued to transmit all critical data directly to the BS.

Mathematical analysis

In this section was calculated the expected delay of *CH* selection process after a wind turbine became faulty. Also, expected duration of each round to collect fault decisions by the *CH* from a number of critical nodes was analyzed.

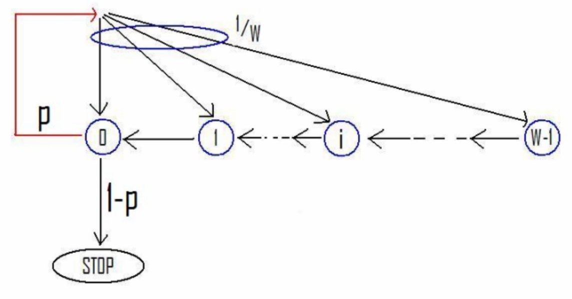

Fig. 2. Back-off process during *CH* selection

Cluster Head Selection: All critical nodes formed a single collision domain. The total number of critical nodes(n) around a faulty wind turbine was $\pi(d_f)^2\lambda$. Each of them turned its receiver on and continuously tried to broadcast *CH-claiming* message using IEEE 802.11 DCF protocol until it heard a successful transmission by any critical node. A node could know whether a given transmission was successful by examining the data it had received. Figure 2 depicts the back-off process of a critical node during *CH* selection process. Before each transmission attempt, a node did back-off and then broadcast its data. It assumed that the average probability of a node's transmission being collided with other node's data was *p*. If the transmission failed (which has the probability of *p*) then it again did back-off and then broadcast. This back-off and broadcast sequence was repeated until any node was successful.

P_i was the steady state probability that a particular node remained at back-off slot *i* ($0 \leq i < w$ where *w* was the minimum contention window length). τ was the probability that a node transmitted in a given back-off slot.

For $i > 0$, $P_i = P_0 \times p \times \frac{1}{w}(w-i)$

Now, $P_0 + \sum_{i=1}^{w-1} P_i = 1$

$$\Rightarrow P_0 + p \times P_0 \times \sum_{i=1}^{w-1}(1 - \frac{i}{w}) = 1$$

$$\Rightarrow P_0 + p \times P_0 \times [(w-1) - \frac{1}{w} \times \frac{(w-1)w}{2}] = 1$$

$$\Rightarrow P_0 + p \times P_0 \times (w-1)/2 = 1$$

$$\Rightarrow P_0 = \frac{1}{1 + p(w-1)/2} = \frac{2}{2 + p(w-1)}$$

Probability of transmission by a node in a given back-off slot,

$$\tau = P_0 = \frac{2}{2 + p(w-1)} \text{ .... } (1)$$

When a node transmitted in a given back-off slot, the probability that none of remaining *n* - 1 nodes transmitted in that slot was $(1-\tau)^{n-1}$. So, the probability of collision as seen by the transmitting node was

$$p = 1 - (1-\tau)^{n-1} \text{ .... } (2)$$

It can solve (1) and (2) numerically to get the values of τ and *p*.

A particular back-off slot was empty with probability $P_{NT} = (1-\tau)^n$, contained a successful transmission with probability $P_S = n\tau(1-\tau)^{n-1}$ or contained a collision with probability $P_C = 1 - P_{NT} - P_S$. When any node successfully broadcast a data packet, the *CH* was selected. If σ was the length of the empty back-off slot then before the first successful transmission, the average slot length $\sigma_{avg} = \sigma \times P_{NT}/(1-P_S) + T_C \times P_C/(1-P_S)$. Here, T_C = *data-time* + *DIFS* and it was the time duration of a collision. After Simplification, it forms

$$\sigma_{avg} = T_C - (T_C - \sigma)\frac{(1-\tau)^n}{1 - n\tau(1-\tau)^{n-1}} \text{ .... } (3)$$

Every time when a packet was transmitted, on an average, the node had to wait (*w* + 1)/2 number of back-off slots before transmission. If the transmission failed (probability was *p*) then the node repeated the back-off process and transmitted again. Probability of successful transmission at i^{th} attempt was p^{i-1}(1-*p*) and the node had to wait a total of *i*(*w* + 1)/2 back-off slots before successful transmission in this case. If the node successfully transmitted within at most *m* attempts then expected total delay of the first successful transmission by any node was

$$D = \sum_{i=1}^{m} i \times \frac{w+1}{2} \times \sigma_{avg} \times p^{i-1}(1-p)$$

$$= (1-p)\frac{w+1}{2}\sigma_{avg} \times \sum_{i=1}^{m} ip^{i-1}.$$

The probability of success of a transmitted packet was 1 - *p*. So, *m = 1/(1-p)* which implies

$$D = \frac{w+1}{2(1-p)}(1 - 2p^{\frac{1}{1-p}})\sigma_{avg} \text{ .... } (4)$$

Thus, the *CH* would be selected within *D* seconds after the cluster formation process started.

Fig. 3 shows *CH* selection delay with different number of candidate nodes, *n* under different initial contention window sizes, *w*. *CH claiming* message length was 510μ*s* and physical layer parameters were taken from IEEE 802.11b standard (σ = 20*µs*, *SIFS* = 10*µs*, *DIFS* = 50*µs*). The maximum tolerable delay was application dependent. Suppose that in a network topology, 20 nodes around a faulty wind turbine can sense the fault. If the delay constraint was 12*ms* then can choose either *w* = 32 or *w* = 64 (Figure 3). But if it was 16*ms* then choose *w* = 16 also. In this way, system can select appropriate parameters of transmission based on *CH* selection delay requirement.

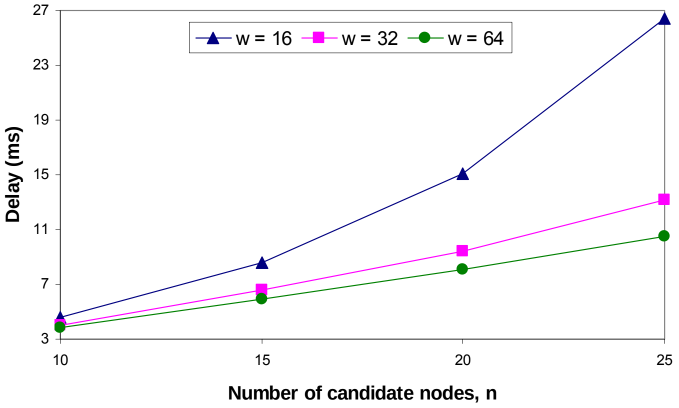

Fig. 3. CH selection delay at different number of candidate nodes

Round Duration: Let us assume that *n* nodes tried to send positive data to the *CH*. Each of them listened to the medium. A node, waiting to transmit its decision to the *CH*, stopped transmission attempt as soon as it found *k* number of successful decisions sent by other nodes. Every time a decision was successfully broadcast, number of transmitting nodes remaining was reduced by 1. Let us assume that $\tau(n)$, $p(n)$ and $\sigma_{avg}(n)$ denoted respectively the values of τ, p and σ_{avg} (subsection 3.1) when the total number of nodes in the collision domain was *n*. The 1st broadcast was successful within $t_1 = \frac{w+1}{2[1-p(n)]} \times [1-2p(n)^{\frac{1}{1-p(n)}}] \times \sigma_{avg}(n)$ seconds. Now the successful node stopped and remaining *n*-1 nodes competed. The 2nd broadcast was successful within $t_2 = \frac{w+1}{2[1-p(n-1)]} \times [1-2p(n-1)^{\frac{1}{1-p(n-1)}}] \times \sigma_{avg}(n-1)$ seconds after the 1st successful broadcast. Actually, the 2nd successful broadcast required less time than t_2 because the competing nodes were already in the middle of back-off process. Similarly, the *k*th successful broadcast occurred within $t_k = \frac{w+1}{2} \times [1-2p(n-k+1)^{\frac{1}{1-p(n-k+1)}}] \times \frac{\sigma_{avg}(n-k+1)}{1-p(n-k+1)}$ seconds after the *(k-1)*th successful broadcast. So, the *CH* received *k* number of packets in *T* seconds, where

$$T = \frac{w+1}{2} \times \sum_{i=1}^{k} [1-2p(n-i+1)^{\frac{1}{1-p(n-i+1)}}] \times \frac{\sigma_{avg}(n-i+1)}{1-p(n-i+1)} \text{ .... } (5)$$

At the beginning of each round, the *CH* broadcast a *round initiation* message which was sent collision free because no other nodes in the cluster sent until this packet was broadcast. Assume that this packet needed T_1 seconds for its transmission and ignored propagation delay. The length of each round would be greater than $T + T_1$ seconds because the *CH* had to compete (using frequency band f_s) with non-cluster nodes to send its decision to the BS. From (5), it showed $T = 17.59ms$ when $n = 10$, $k = 5$ and $w = 16$, and $T_1 = 510\mu s$ in our experiments (Result and Discussion Section). Therefore, each round had a length nearly equal to 0.018*s*. If the application demands smaller round length then it can change the values of some or all of *w*, *n* and *k* in order to obtain the correct round length. The section that described Cluster Operation presented the way to change the value of *n*.

Results and Discussion

The proposed scheme was compared with the non-clustered version of the work presented by Angeles and Bouabdallah, 2009 which supported both data gathering and event monitoring simultaneously. The non-clustered version works without cluster formation except the event cluster. For simplicity in referring this version, it's denoted this scheme as *RefPro* (reference protocol). In *RefPro*, *CH* nodes sent their data directly to the BS. In proposed method, a node only sent data directly to the BS when it found that the deadline of a critical packet was near the end. The regular data packets had no deadline at all. The system ran simulations using NS-2 (NS-2 website, 2006) network simulator. 20 wind turbines was pleased in a grid structure having a 1.6*km* × 1.2*km* area. Distance between 2 successive wind turbines on a grid was 400*m*. 221 number of sensor nodes also placed on the same grid structure. Distance between 2 successive nodes on a grid was 100*m*. Figure 4 shows the wind farm with deployed sensor nodes and the BS. Each intersection of the grids contained a sensor node. Locations of wind turbines are shown by solid filled circles. It's employed 1 Mbps IEEE 802.11 DCF MAC protocol. In All experiments, data length was 510μs, ACK length was 304μs and physical layer parameters were taken from IEEE 802.11b standard (σ = 20μs, *SIFS* = 10μs, *DIFS* = 50μs). Proposed system considered a total of 3 simultaneous faulty wind turbines in the farm (in Fig. 4, faulty turbines are marked as F_1, F_2 and F_3). The nodes around the faulty turbines generated critical data while other nodes generated regular data. Each *CH* node sent 10 *pps* (packets/second). The proposed scheme was compared with *RefPro* in terms of end-to-end success rates of critical and regular data. Compared them in terms of the lifetimes of the networks and also did experiments with different positions of faulty turbines. Data obtained shows in the case considered in Fig. 4. All other results had similar comparative performance. When a wind turbine became faulty, any node within 200*m* could sense it.

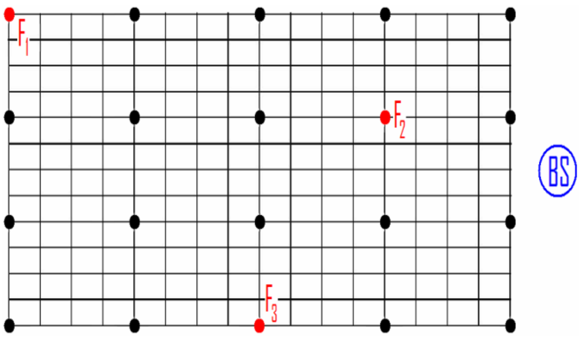

Fig. 4. Locations of faulty wind turbines and the BS

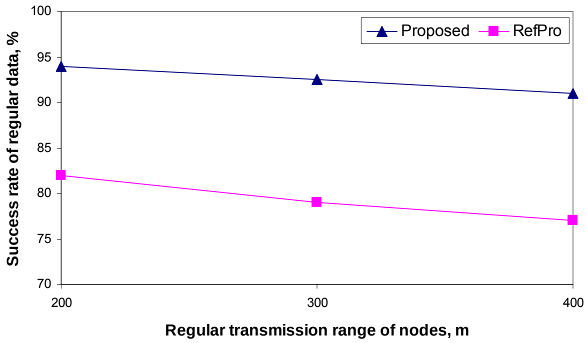

Fig. 5. Success rate of regular data at different regular transmission ranges of nodes (*CH* data rate = 10.0 *pps*, regular data rate = 1.0 *pps*)

Fig. 5 shows success rate of regular data at different regular transmission ranges (d_s) of nodes. *CH* data rate was 10 *pps* and regular data rate was 1 *pps*. The regular transmission ranges did not apply to *CH* nodes in *RefPro* and also did not apply to any node in proposed method when it found a critical packet's deadline near its end. As all *CH* nodes in *RefPro* sent critical data with high transmission power, a large number of regular data transmissions were collided with critical data transmissions. In proposed method, a node used high transmission power only when it was necessary to meet the deadlines of critical data, and the number of such transmissions was low indeed. That was why regular data success rates were much (15% - 18%) higher in proposed method than those found in *RefPro*. This gain was important as it wanted to get as many regular data as possible for statistical analysis at later stages. As transmission range of the nodes increased, total number of nodes inside collision domain of a node also increased and caused higher collision probability. Therefore, data success rate decreased in both methods as transmission range increased.

Fig. 6 shows success rate of regular data at its different generation rates. *CH* data rate was 10 *pps* and regular transmission range was 300*m*. Increased rate of regular data in the network caused higher amount of competition among the nodes to capture the shared medium. This leaded to higher number of collisions which caused lower success rate of the regular data in both methods. However, due to lower number of high power transmissions, the proposed method performed better than *RefPro*. Indeed, it had 8% to 17% higher success rate of regular data.

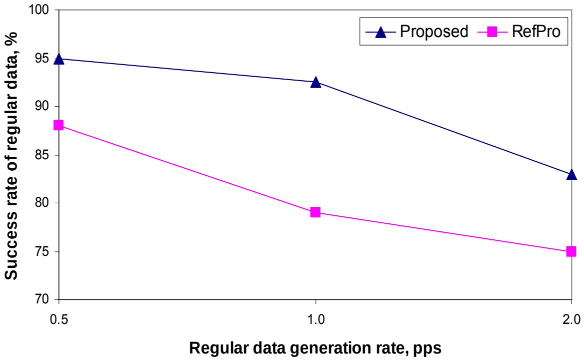

Fig. 6. Success rate of regular data at its different generation rates (*CH* data rate = 10.0 *pps*, regular transmission range = 300.0 m).

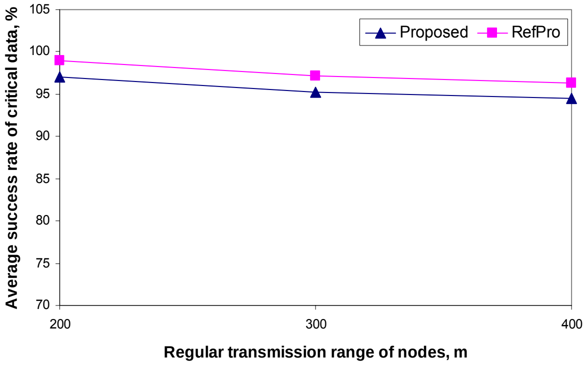

Fig. 7. Average success rate of critical data at different regular transmission ranges of nodes (*CH* data rate = 10.0 *pps*, regular data rate = 1.0 *pps*).

Average success rates of critical data at different regular transmission ranges (d_s) of nodes and at different generation rates of regular data are shown in Fig. 7 and Fig. 8 respectively. The figures show the average of data about all three simultaneous faulty wind turbines. System limit the deadline of each critical packet to 0.1*s* and regular data has no deadline. When a critical packet did reach the BS within 0.1*s* after its generation at the source node, it was accepted as meaningful. Otherwise, it was discarded as unsuccessful and did not care about average delivery delay of critical data. Rather, were concerned about the required deadline set for the wind farm monitoring application. As critical data were sent directly by *CH*s to the BS in *RefPro*, it had a higher success rate than the proposed method. But it outperformed the proposed method merely by at most 2%, and the proposed method had a success rate of above 94% in all cases. Above 90% success rate was acceptable because 9 out of 10 critical packets sent render sufficient information about the ongoing fault. On the other hand, transmission using high power in *RefPro* caused lower success rate of regular data (Fig. 5 and Fig. 6) and most importantly, lifetime of the network was significantly lower in *RefPro* compared to the proposed method (please see Figure 10 and Fig. 11). The success rates in both methods gradually decreased with the increase in regular transmission range of nodes or with the increase in regular data generation rate. The reason was same that caused similar behaviour in Fig. 5 and Fig. 6.

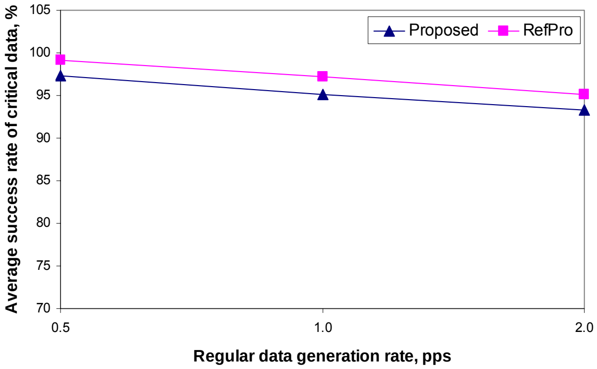

Fig. 8. Average success rate of critical data at different generation rates of regular data (*CH* data rate = 10.0 *pps*, regular transmission range=300.0 m)

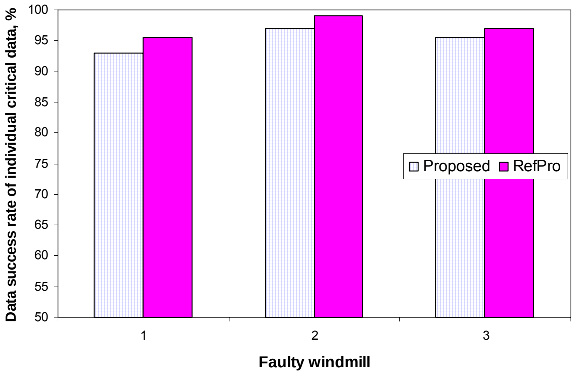

Fig. 9. Success rate of individual critical data (*CH* data rate = 10.0 *pps*, regular data rate = 1.0 *pps*, regular transmission range = 300.0 m)

Fig. 9 shows success rate of critical data about individual faulty wind turbines. *CH* data rate was 10 *pps*, regular data rate was 1 *pps* and regular transmission range of nodes was 300*m*. In both methods, individual critical data success rates were close to one another, i.e., all ongoing faulty wind turbines were remarkably detected.

Maximum energy spent by any node in the network at different regular transmission ranges (d_s) of nodes and at different generation rates of regular data are shown in Fig. 10 and Fig. 11 respectively. This parameter was very crucial because it would reflect the lifetime of the network. The smaller the maximum node energy spent the larger the network lifetime. As sensor nodes were powered by non-replaceable batteries, it's want to have lifetime as long as possible. Fig. 10 shows that the maximum node energy spent in proposed method was at least 73% less than that found in *RefPro*. Fig. 11 shows an energy saving of at least 75% in proposed method. With the increase in regular transmission range (d_s) of nodes, the maximum node energy spent reduced in proposed method as shown in Figure 10. Basically, maximum energy dissipation occurred in one of the nodes that were very close to the BS. When transmission range increased, more number of nodes sent data directly to the BS. These direct transmissions reduced communication burden of nodes close to the BS. This is why maximum node energy dissipation reduced in proposed method. But in *RefPro*, *CH*s sent data directly to BS using high power and therefore, maximum node energy was used in any of the *CH*s. As a result, maximum node energy spent in the network did not change noticeably with increased transmission range of other nodes in *RefPro*. When regular transmission range was kept fixed and regular data generation rate increased, the average number of collisions per transmission also increased. As a result, more number of retransmissions became necessary which caused higher energy consumption. Therefore, Fig. 11 shows that maximum node energy used in both methods increased with increase in regular data generation rate.

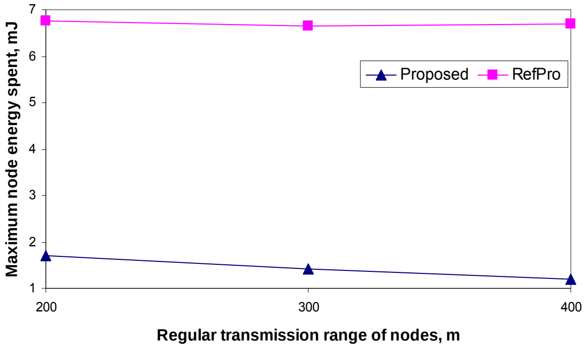

Fig. 10. Maximum node energy used at different regular transmission ranges of nodes (*CH* data rate = 10.0 *pps*, regular data rate = 1.0 *pps*)

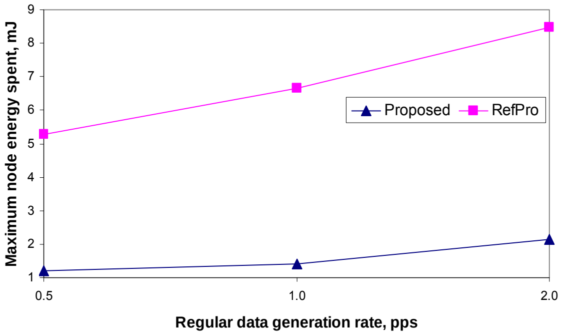

Fig. 11. Maximum node energy used at different generation rates of regular data (*CH* data rate = 10.0 *pps*, regular transmission range = 300.0 m)

Conclusion

The proposed system presented a novel method applicable to sound data collection as well as fault monitoring within the same WSN deployed in a wind farm. Very few prior works considered both types of applications simultaneously. When a wind turbine became faulty, a cluster around the faulty wind turbine was automatically formed and a cluster head was selected to aggregate the data of fault sensing nodes. The aggregated information was then transferred from the cluster head to the base station through multi-hop communication. Inside the cluster, a different transmission channel was utilized in order to avoid interference with other transmissions occurred outside of the cluster. The work coped with multiple simultaneous faulty wind turbines. The proposed method compared to existing similar works by performing extensive simulation. The proposed method outperformed the existing methods in terms of periodic sound data collection and energy conservation. Due to lower number of high power transmissions, the proposed method had a regular data success rate 8-17% higher than existing methods. In terms of timely faults detection and notification, this method had a 94% data success rate which could be compared to the existing methods and it also enhanced network lifetime by 73-75% which was a significant improvement. The proposed method did not make any compression of periodic sound data. Future work would consider efficient techniques to compress sound data in order to improve the performance further.

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. 2002. A Survey on Sensor Networks. IEEE Communications Magazine: 102-114.

Angeles, M.E.R. and Bouabdallah, N. 2009. Event reporting on continuous monitoring wireless sensor networks. Proceddings of IEEE GLOBECOM: 1–6.

Banerjee, T., Xie, B. and Agrawal, D. P. 2008. Fault tolerant multiple event detection in a wireless sensor network. Journal of Parallel and Distributed Computing. 68: 1222–1234.

Biagioni, E. S. and Sasaki, G. 2003. Wireless sensor placement for reliable and efficient data collection. Proceedings of the 36th Hawaii International Conference on System Sciences, Hawaii.

Colby, W. D. *et al*. 2009. Wind turbine sound and health effects: an expert panel review. American Wind Energy Association, Canadian Wind Energy Association.

IEEE Std 802.11™-2007. 2007. IEEE , 3 Park Avenue, New York, USA.

Li, M., Liu, Y. and Chen, L. 2008. Nonthreshold-based event detection for 3D environment monitoring in sensor networks. IEEE Transactions on Knowledge and Data Engineering. 20(12): 1699–1711.

Ma, Y. *et al*. 2008. Air pollution monitoring and mining based on sensor grid in London. Sensors 8: 3601–3623.

Mainwaring, A. *et al*. 2002. Wireless sensor networks for habitat monitoring. Proceedings of the 1st ACM Int. Workshop on Wireless Sensor Networks and Applications, New York, USA.

Moorhouse, A., Waddington, D. and Adams, M. 2005. Proposed criteria for the assessment of low frequency noise disturbance. University of Salford, UK.

NS-2 website, URL http://www-mash.cs.berkeley.edu/ns/.

Pottie, G. 1998. Wireless Sensor Networks. Proceedings of Information Theory Workshop. San Diego, CA, USA: 139-140.

Rogers, A.L., Manwell, J.F. and Wright, S. 2006. Wind turbine acoustic noise: a white paper. University of Massachusetts at Amherst.

Sonus Pty Ltd. 2010. Wind farm technical paper, environmental noise. 17 Ruthven Avenue, ADELAIDE, SA, Australia. www.sonus.com.au.

Xu, X., Luo, J. and Zhang, Q. 2010. Delay tolerant event collection in sensor networks with mobile sink. Proceedings of IEEE of INFOCOM (2010).

Zhu, Y. and Ni, L.M. 2008. Probabilistic approach to provisioning guaranteed QoS for distributed event detection. Proceedings of IEEE INFOCOM (2010).