ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A SKEWED SLOT SCHEDULING APPROACH FOR REAL-TIME DATA TRANSMISSION IN INDUSTRIAL WIRELESS SENSOR NETWORKS

Md Abul Kalam Azad*, Md Abdur Rahman, and Amina Khatun
* Department of CSE, Jahangirnagar University, Dhaka-1342, Bangladesh
Department of Mathematics, Jahangirnagar University, Dhaka-1342, Bangladesh
Department of CSE, Jahangirnagar University, Dhaka-1342, Bangladesh

DOI: 10.5281/zenodo.439212

ABSTRACT

Improving the data transmission reliability of a time-constraint monitoring and control application in industry is a challenging task. Contemporary solutions are based on either Time Division Multiple Access (TDMA) or the hybrid model of TDMA and Carrier Sense Multiple Access (CSMA). However, most of them are burdened with a complex slot allocation scheme while others are vulnerable to the time varying nature of wireless links. Therefore, we propose a skewed slot scheduling approach that maintains a global time constraint in data delivery using a CSMA-base data transmission within a big-slot. All nodes at the same tree depth collectively allocate a big-slot and share it for data transmission reliability. The new approach is not only simple in design but also highly dependable against dynamically changing network topology. The approach also maximizes data aggregation and filtering, thus improving the balancing of energy consumption among sensor nodes.

KEYWORDS: Hybrid architecture, monitoring and control, real-time system, reliable transmission, slot scheduling.

INTRODUCTION

Generally, a safety-critical industrial application requires the strict time-constrained and reliable data transmission in Wireless Sensor Networks (WSNs) [1]. However, the industrial environment is often non-friendly to wireless communication due to the dynamic changing of network topology by the various level of interferences. If a protocol does not respond to them in time, the safety of observed entities may be in danger. Moreover, the replacement of the battery cannot be often trivial because the sensor nodes may be placed on high shelves or in hazardous places. Therefore, the protocol should address the reliability requirement of applications as well as the balancing of power consumption among the nodes to prolong the lifetime of WSNs.

Z-MAC [2] incorporates CSMA into TDMA to improve channel utilization of TDMA under low channel utilization. It applies DRAND [3] to do slot assignment in which no two nodes within a two-hop communication neighborhood are assigned the same slot. By combining CSMA with TDMA, a node can achieve efficient channel utilization in low traffic by stealing an unused slot. The priority based contention process maximizes the channel utilization indeed. However, maximizing the channel utilization does not necessarily maximize data delivery [4]. Moreover, the contention process incurs more delay and energy consumptions. To overcome them, TreeMAC [4] devises a time slot assignment algorithm depending on the depth and parent-child relationship of a tree structure. TreeMAC assigns non-overlapping frames to all nodes and allocates a transmittable slot to each according to depth. In this way, it not only removes the funneling effect [5] at congested nodes but also reduces delay. However, the physical transmission range based topology for slot reuse tends to incur an irregular interference among nodes. It also bypasses an energy efficient tree construction and hinders the aggregation of data packets.

Toward having a flexible real-time system, WirelessHART [6] standards are developed over IEEE 802.15.4 PHY [7]. Using a fixed 10ms time slot, WirelessHART uses TDMA with frequency hopping for channel access in 2.4 GHz band. The standard uses 16 channels in the range of 11-25. It adopts a slotted hopping scheme where the channel is changed every slot. Moreover, it supports blacklisting for eliminating interfering channels. The core functionality of WirelessHART is dependent on a network manager that requires multi-channel hardware support.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

However, it may not be a viable solution for low cost and low configurable WSNs. Instead, I-MAC [1] uses a single channel and assigns unique time slots to each node. Children send slot demand to their parent, and in response, the parent assigns the non-overlapping slots. Additionally, I-MAC adds some special functionality such as bi-directional tree construction and spare time utilization scheme (STUS) for enhancing reliability. However, the protocol is intended to a relatively stable network where a failed link is not recovered until the data acquisition ratio falls below a threshold value. The lost packet is salvaged through STUS. However, giving more chances to a failed node does not ensure forwarding a data packet to its parent; rather it only wastes precious energy.

For industrial automation using WSNs, GinMAC [8] has evolved with three remarkable features: Firstly, it runs an off-line dimensioning process, which defines application traffic, channel characteristics and a tree topology. Secondly, a TDMA scheduler assigns exclusive transmission slots to each node within a fixed epoch length. Thirdly, based on observed channel characteristics, redundant transmission slots are added in a frame for reliability control without violating the delay bound of an epoch. As GinMAC calculates slot demand on the basis of pre-assumed channel characteristics and network parameters, the network performance will be surely degrade when deployed in a dynamically changing environment.

A graph based protocol [9] executes a centralized link scheduling algorithm based on interference modeling. It optimizes link scheduling delay by linear programming and maximum likelihood function. On the contrary, the S-Web based MAC [10] follows the checkerboard and dartboard based slot scheduling to reduce the number of nodes contending for a media at the same time. In RNP [11], a gateway initially defines an arbitrary schedule for nodes, and later on it can reschedule the transmission to adjust the changes in network by beacon messages. Each frame in RNP begins with a TDMA phase, which is followed by either another TDMA or a CSMA phase depending on the number of packet drops at previous phase. These categories of protocols try to find out an efficient scheduling algorithm with minimum number of slots. However, using minimum number of slots does not always justify the optimum performance as long as the superframe length is remained below the delay constraint of intended applications.

DMAC [12] schedules time slots in a staggering style in which each node skews its wake-up time ahead of the sink's schedule according to its depth in a data gathering tree. However, WIRES [13] schedules time slots based on the rank of nodes. The static slot allocation in WIRES is designed to let nodes have enough time to transmit their data, and go to sleep for the rest of time. On the other hand, TDGEE [14] starts slot scheduling from the leaf node of a tree. The immediate parent in return assigns the required time slots, and slot assigning process progresses toward a sink. IH-MAC [15] protocol, however, achieves high channel utilization through using both broadcast scheduling and link scheduling that dynamically switches between them depending on load. Through various slot assigning strategies, these protocols try to reduce end-to-end delay. However, without optimized slot length and proportional amount of time slots, these protocols cannot be justified for industrial applications.

The main drawbacks of the existing MAC protocols are that they are not enough flexible and adaptable to the dynamicity of industrial WSNs. An example of dynamicity is that a transient topological change may happen in static WSNs due to time varying wireless links. This behavior demands for a new slot scheduling in industrial arena. Therefore, to overcome the shortcomings of the contemporary MAC protocols for real-time and reliable data communication, we propose a hybrid MAC termed as *BigMAC* that uses CSMA within a big time slot for data transmission reliability. BigMAC is simple in design yet robust enough in scalability, transitive-link, mobility, and energy usage. In short, the major advantages of BigMAC are follows:

- Instead of a dedicated tree, we use a table-based logical tree where a node can access an alternate parent [16], if there is any.
- We design big-slot concept based on skewed time waiting function [17], which quantitatively represents the nature of data flow in tree-based WSNs.
- In here, TDMA is applied network wide to support real-time operation, while CSMA is applied locally (between parent and child) to optimize contention level and to alleviate hidden terminal problem.

The rest of the paper is organized as follows: In section II, we depict the problem findings and motivations behind of BigMAC. In section III, we elaborately express the design principle of BigMAC. In section IV, we implement BigMAC in a network simulator and evaluate its performance. Finally, we draw some concluding remarks about the paper in section V.

BACKGROUND

Motivations and Problem Statements

TDMA eliminates contention among the nodes and gives them a guaranteed chance to deliver data packet to a sink within a specified time. To realize this, a set of unique slots are assigned to each node according to its demand

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

before a data session begins. Data transmission rate is reasonably good when the wireless link is stable. However, problem arises if one or more nodes lose their links to a sink. To overcome this problem, we can perform tree reconstruction and slot rescheduling after some percentage of nodes fails to deliver data packets to a sink. In this case, some amount of slots will be wasted. Consider a simple sensor network of 13 nodes as in Fig. 2 as a part of whole monitoring and control system in Fig. 1. Suppose that link (1, 2) is broken. The slots allocated to nodes 2 and 4 are wasted until tree reconstruction and slot rescheduling are made. Some MAC protocols such as TreeMAC and I-MAC require slot rescheduling. However, if some slot scheduling allows node 2 to change its parent to node 7 and still to use the same slot, the shortcomings such as control overhead and slot wastage can be resolved.

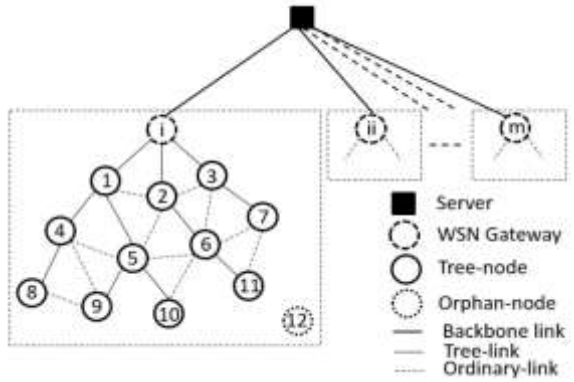

Fig. 1. A network model

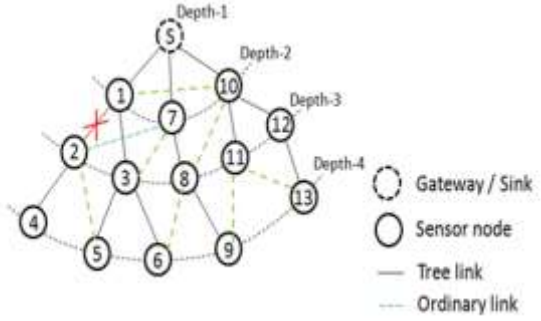

Fig. 2. An example of tree maintenance.

Reusing of time slots is another issue. It is very difficult to identify whether or not any two nodes can use the same slot since the interference range is always farther than a physical transmission range [18]. In fact, TreeMAC and WIRES that reuse slots are not completely free from interference. Meanwhile, I-MAC does not reuse slots; however, it requires much more slots if the network size increases. We can achieve channel efficiency and timely delivery by assigning a shared big-slot to the nodes at same depth. Then, all nodes at the same depth compete for acquiring a channel within the same big-slot, limiting the delivery time and allowing parallel transmission opportunistically. For example, node 4 and node 13 can transmit data packet within the identical time slot if they do not interfere with each other.

Another motivation factor is that only a small portion of TDMA time slot is used for data transmission. In TDMA based monitoring and control applications, 100 bytes of data packet and 20ms of time slot are well accepted values [19]. In fact, these irrational values would cause a lot of channel inefficiency. For example, IEEE 802.15.4 supports 256 Kbps data rate. It will take about 3.125ms to send one data packet of 100 Bytes with no interference. The remaining time is either totally wasted or partially used for the exchange of control messages.

To address above motivation factors, we cancel out the idea of allocating distinct time slots to each node. Instead, we allocate a big slot to all nodes at same tree depth so that the nodes send their data packets in a contention-based manner within the time slot. This simple relaxation in slot allocation makes the parallel transmission possible opportunistically such that any two nodes can send packets simultaneously if they do not interfere each other. This would obviously enhance network efficiency.

Notations and Definitions

For convenience, we use some notations and definitions as follows:

depth(i): The depth of node i
N(i): A set of neighbors of node i
C(i): A set of children of node i
P(i): The parent of node i

Definition 1: A big-slot is a time span that all nodes at the same depth share to receive data packets from their children and transmit their data packets to their respective parents using CSMA technique. A big-slot allocated to the nodes at depth i is denoted as BS(i), the portion of the BS(i) for the nodes at depth i to receive data packets from their respective children is denoted by $BS^{RX}(i)$ and that of the BS(i) for the nodes at depth i to transmit data packets to their respective parent is denoted as $BS^{TX}(i)$.

Definition 2: A super frame (SF) consists of the aggregated big-slots used in a tree structure.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

$$SF = \frac{1}{2} * \sum_{k=1}^{H} BS(k) \qquad (1)$$

The summation is divided by ½, because of the half-way overlapping of sending and receiving big-slot in the data transmission process.

DESGIN PRINCIPLE OF BigMAC PROTOCOL

Protocol Structure

The protocol structure consists of an Initial Construction Phase (ICP) and a repeating cycle that includes a Reliable Data Transmission Period (RDTP) and a Maintenance Period (MP). RDTP works in a contention-based mode with time constraint. MP consists of three optional maintenance tasks such as time synchronization, tree construction, and slot scheduling.

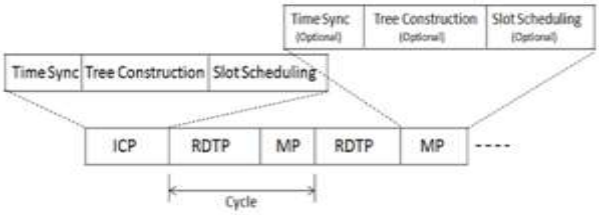

Fig. 3. Protocol structure.

During ICP, initial time synchronization, tree construction, and slot scheduling are performed as shown in Fig. 3. In slot scheduling, a unique big-slot is assigned to a collection of nodes at each tree depth. The big-slot is used to maintain the time constraint and all nodes at the same depth use CSMA within a big-slot.

Tree Construction and Maintenance

Link quality estimation

It is of great importance to have reliable tree-links in building a tree topology since they carry data packets in monitoring and control applications. Thus, we need to identify bi-directionally reliable tree-links while building a tree. However, it is not easy to identify a reliable link because link quality varies unpredictably according to time and space [20].

One popular method is to measure the physical characteristics of received packets *e.g.*, Received Signal Strength Indicator (RSSI) and Link Quality Indication (LQI). However, each individual metric may not classify link quality for the entire spectrum accurately. Recently, an experiment is done using TelosB motes running TinyOS 2.1 in an indoor office [1] and evaluated the results with a joint metric, *linkq*, used in paper [21].

$$linkq = \sqrt{\overline{RSSI_w}^2 + \overline{LQI_w}^2} \qquad (2)$$

where, $\overline{RSSI_w} = \frac{\sum_{k=1}^{m}(RSSI_k + 100)}{n}$, $and\ \overline{LQI_w} = \frac{\sum_{k=1}^{m} LQI_k}{n}$, m is the number of received packets, and n is the number of transmitted packets ($0 < m \leq n$). Using the *linkq*, we can judge a link to be reliable if it is greater than a specified threshold constant, $R_{Link\_Threshold}$.

Bi-directional reliable link

We define link (a, b) to be bi-directionally reliable (B-reliable) *iff* two directional links from a to b and from b to a are reliable. We construct a tree with bi-directionally reliable tree links by modifying the tree construction method.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

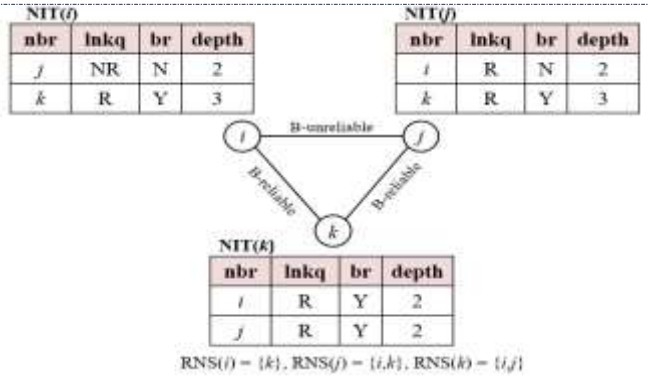

Fig. 4. Illustration of determining B-reliable link.

As shown in Fig. 4, every node, *i* maintains its neighbor information table,

$$NIT(i) = \{(x, lnkq(x), br(x), depth(x)) | x \in N(i)\} \qquad (3)$$

where *N(i)* denotes a neighbor set of node *i*, *lnkq*(*x*) indicates whether link (*i, x*) is reliable (R) or not reliable (NR), and br(*x*) indicates whether link (*i, x*) is B-reliable or not. Let us define a *reliable neighbor set* {*RNS(i)*} as a set of neighbors who are reliable to node *i*. The node *i* that has obtained the *RNS(j)* from node *j* determines link (*i, j*) to be *B-reliable* if entry *j* of *NIT(i) is R* and *i* ∈ *RNS*(*j*). Therefore every node *i* is required to include *RNS(i)* in a control message. Node *i* updates its *N(i)* and *NIT(i)* whenever it receives or overhears any control message from its neighbor. If node *i* does not receive or overhear any control message from a neighbor within a specified time bound, *NBR_timeout*, it removes the neighbor from *N*(*i*) and *NIT*(*i*).

Reliable tree construction

To enhance reliability, we construct a tree that consists of the B-reliable links only. A sink starts the protocol operation by issuing a *SYNC* message for time synchronization. Upon receiving *SYNC*, every node performs initial time synchronization and rebroadcasts *SYNC*. We employ the modified Flooding Time Synchronization Protocol (FTSP) [1]. During this process, every node gathers link quality information as an initial *NIT* for its neighbors.

After initial time synchronization, a sink is the only tree member in a network and other nodes are just orphan nodes. A sink initiates tree construction by issuing a Tree Construction Request message, TCR = (*sinkID*, W_1, *a*, *RNS*). Upon receiving TCR, an orphan node that has a reliable link joins to sink by sending a Join Request message, JREQ = (*sender*, *receiver*, *depth*, *RNS*). Upon receiving JREQ, a member sends a Join Response message, JRES = (*sender*, *receiver*, *depth*, W_1, *a*, *RNS*) and takes the orphan as its child if the corresponding link between them is B-reliable. When an orphan receives JRES, it takes the member as its parent. Another orphan who has overheard JREQ can take the same procedure to become a member if its link is B-reliable. If an orphan overhears multiple JREQs from different members with B-reliable links, it pairs with the member that has the shortest distance (depth) to the sink. To prevent the collision among JREQ's during tree construction, each node sets a waiting timer before issuing JREQ as in [17].

Wait Time Generation Function

In WSNs operation, the waiting time function contains two basic principles: *Firstly*, a node must wait for all of its children completing their data transmission. This way of skewed waiting time is essential for improving data packet aggregation. *Secondly*, the waiting time gap between any two nodes of two consecutive depths should increase exponentially as depth decreases. The reason is that the nodes at lower depth have higher contention to acquire channel since the possibility of parallel transmission decreases while the total size of data packet for all nodes at each depth is kept almost the same. The wait time distribution function is given below:

$$WTime(d) = W_1 \times a^{d-1} \qquad (4)$$

where *WTime*(*d*) is the time that a node at depth *d* has to wait to transmit data packets to its parent and the range of the base a is in (0, 1]. Since a sink does not have to send any packets, *WTime*(1) (= W_1) is equal to SF.

Cycle Time and Big-Slot Length

A superframe W_1 should be greater than the summation of transmission times of all packets from each node to a sink, assuming that each node generates only one packet within a superframe. Thus, the bound of W_1 can be given as follows:

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

$$\sum_{d=2}^{H}(d-1)*n_d*T \le W_1 \le \sum_{d=2}^{H}(d-1)*n_d*E[D] \qquad (5)$$

where H is the depth of a tree, n_d is the number of nodes at depth d, T is the one-hop transmission time of a packet and E[D] indicates an expected packet delay that a node needs to send a packet to its parent successfully using CSMA. If an application deadline permits, we can take a value larger than this to relax the time constraint of data delivery process. We derive the size of the BS of a node at depth i as follows:

$$BS(i) = BS^{Rx}(i) + BS^{Tx}(i)$$
$$\begin{cases} W_1(a^{-2}-1)*a^i & if\ a\ node\ is\ an\ intern\ node \\ W_1(a^{-2}-a^{-1})*a^i & if\ a\ node\ is\ a\ leaf\ node \\ W_1(a^{-1}-1)*a^i & if\ a\ node\ is\ a\ sink\ node \end{cases} \qquad (6)$$

The BS size depends on the depth of a node in a tree topology. As the value of a is inversely related to the children spawning rate, its value is always less than 1. Thus, we can say that the lower depth node requires larger BS than that of the higher depth node according to (6). It is worth mentioning that the summation of all BS's is larger than the cycle time as the adjacent BS's are always overlapped with each other.

Big-Slot Assignment Algorithm

Once time synchronization is finished, each node performs slot scheduling in a totally distributed manner using (6). Let us denote the start time of a cycle by *sTime*. Then, assuming that time is perfectly synchronized, a node at depth i has the following big-slot schedule:

$$RxTime(i) = sTime + WTime(i+1) \qquad (7)$$
$$TxTime(i) = sTime + WTime(i) \qquad (8)$$
$$SleepTime(i) = sTime + WTime(i-1) \qquad (9)$$

where *sTime* = GT(0) + MaxICP, and GT(0) is a global time observed at a sink node.

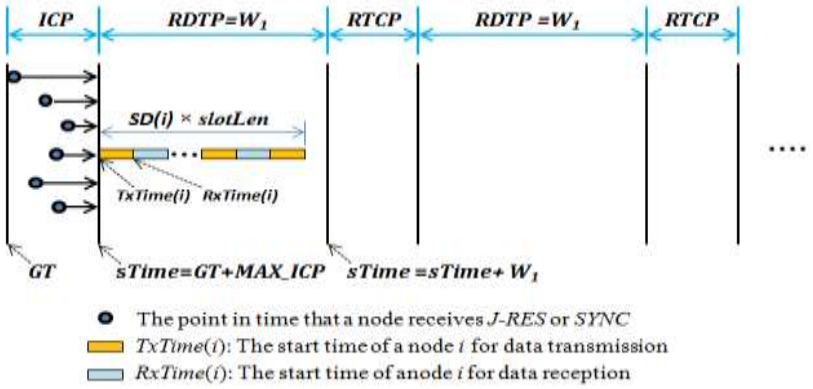

Fig. 5. Timing chart for data transmission algorithm.

Fig. 5 depicts the timing sequence for data transmission. BigMAC starts operation by initiating a tree construction process at Global Time (GT). Every node finishes its tree construction at sometime within ICP and starts its RDTP at the time of GT(0)+MaxICP, where MaxICP is the maximum possible time span that every node finishes joining a tree (*i.e.*, receives JRES from a tree member), time synchronization, and slot scheduling. Then, every node i can obtain RxTime(i) and TxTime(i) from (7) and (8) since it knows *sTime*.

Table 1. BigMAC packet scheduling algorithm

```
// GT(0) : The global time perceived at the sink node
// MaxICP : Maximum value set for ICP
// MaxMP : Maximum value set for MP
At a node that receives JRES or SYNC: //in case that tree construction or time sync is performed
    remove RDTPtimer; //remove the timer that was previously set
    sTime = GT(0) + MaxICP;
    set RDTPtimer = sTime;
    go to sleep;
```

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

```
// Start of the data transmission cycle
At a node of depth i that RDTPtimer expires:
IF the node has children THEN
        set RxWakeuptimer = RxTime(i); // receive first
        go to sleep;
ELSE
    set TxWakeuptimer = TxTime(i); // transmit only
    go to sleep;
ENDIF

// Receiving data before sending
At a node x of depth i that RxWakeuptimer expires:
    set TxWakeuptimer = TxTime(i); // transmit next
S = C(x); // C(x) is the children set of node x

// If there is no child remaining to send, enter sleep mode
At a node x of depth i that receives a packet from its child v:
S = S – v;
IF S == ϕ THEN
    go to sleep;
ENDIF

// Aggregating data before sending to its parent
At a node of depth i that TxWakeuptimer expires:
    set TxEndtimer = TxTime(i) + BS^TX(i);
aggrPacket = packet-aggregation(); // aggregate all queued packets
// see next section for data transmission
IF(packet-transmission(aggrPacket)) || (TxEndtimer expires) THEN
set MP-timer = sTime + W1;
go to sleep;
ENDIF

// For the consecutive data transmission cycle
At a node of depth i that MP-timer expires:
sTime = sTime + W1 + MaxMP;
set RDTPtimer = sTime;
```

The BigMAC packet scheduling algorithm is given in Table 1. Nodes at depth *i* wake up at time RxTime(*i*) and wait to receive packets from its children. If it receives packets from all its children, it immediately gets into sleep mode. At TxTime(*i*), every node at depth *i* wakes up, aggregates its own packet and all received packets, and then tries to send packets to their respective parents using CSMA. As soon as a sensor node finishes sending its aggregated packet, it immediately gets into sleep mode.

PERFORMANCE EVALUATION

The Properties of BigMAC

We compare the features of different MAC protocols especially designed for real-time and/or reliable data delivery as summarized in Table 2. In the TDMA approach, if a node changes its location in a topology, the protocol has to generate a new slot schedule network wide. However, BigMAC generates one distinct slot for the nodes in each depth and if a node changes its depth, it only use the slot allocated for the nodes at changed depth. Thus, it is very simple and suitable for dynamic networks.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Table 2. Comparison of key real-time MAC protocols.

Characteristics	Z-MAC[2]	TreeMAC[4]	BigMAC
Medium access	TDMA+CSMA	TDMA	TDMA+CSMA
Slot scheduling	Distributed	Global	Distributed
Slot reuse	Yes	Yes	Opportunistically
Reliability mechanism	No	No	Yes
Filtering and Aggregation	No	Almost impossible	High
Responsiveness to dynamic topology	Moderate	Low	High
Bi-directional	No	No	Yes
Network size	No restriction	No restriction	No restriction

TreeMAC employs a slot reuse scheme to increase channel utilization; however, the slot scheduling would generate some unused slots, thus wasting channel. Furthermore, the slot reuse incurs irregular interference since the interference range is farther than the transmission range of radio signal. Node movement increases this interference, thus quickly invalidating the slot schedule. BigMAC reuses slots opportunistically such that different nodes at the same depth can send packets at same time if they do not interfere each other. Therefore, the slot sharing can provide an effective slot reuse.

The reliability mechanism using control messages is needed to check whether data is transmitted successfully or not. However, these overhead corresponds to 40%~75% of the channel capacity if packet size is small [18, 22]. Only BigMAC generates a slot schedule that is effective for data aggregation and thus can reduce a lot of control messages. This will also contribute to energy consumption balancing among the nodes at different depths.

Simulation and Discussion

To evaluate the performance of BigMAC protocol, we use the commercially available *QualNet* simulator version 5.0.2. We compare our proposed protocol with Z-MAC and TreeMAC.

Simulation model

To ease topology dimensioning, we enlarge both simulation area and transmission range to the same extent. Therefore, the output produced by these rationally increased parameters is in congruence with the real network model. By using the mathematical formulas in [23, 24], we get the statistically average values of H as 7. Substituting H=7, T=3.125ms [7], $E[D]$=30ms [17], and node distribution value from [23], we get a theoretical range of W_1 (SF) as 0.25s≤W_1≤2.4s. However, we can set an optimum value for W_1 within this range by simulation.

Table 3. Simulation parameters and values

Parameter	Value
Number of node, n	1 sink & 25 sensor nodes
Dimension, d	100 x 100 (m^2)
Simulation time, T	600s
Waiting time, W_1 (BigMAC)	1.6
Base, *a* (BigMAC)	0.7
Slot size, S	20ms
SF (TreeMAC & Z-MAC)	Topology dependent
Transmission range, *R*	20 m (-25 dBm)
Channel frequency, *Fr*	2.4 GHz
Path loss model	2-ray ground
Sensor energy model	MicaZ
Battery model	Linear
Maximum Tx (MAX_TIMES)	2
Data packet length	100 byte

Simulation scenario

Twenty five sensor nodes are uniformly distributed within the boundary of a simulation area of 100m by 100m, and a sink is placed at the middle-top of the area. A node can communicate other node if they are in mutual transmission range of each other. A node may have a set of children, but must have a primary parent in all protocols

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

and a set of secondary parent in BigMAC only. Each sensor node transmits only one packet of 100 bytes including a header every SF, except, of course, for the sink.

All sensor nodes are static. However, the link conditions are dynamically varied to simulate the attenuation and fading. For small communication range, we use Rician fading model in our protocol. We experiment different network metrics such as packet delivery ratio, energy consumption, and end-to-end delay for various noise levels.

Experimental results

- Packet delivery ratio (PDR)

BigMAC constructs a tree topology using the bi-directional reliable links. The protocol, thereby, enhances the data transmission reliability by avoiding asymmetrical and transient links from a tree topology. Moreover, it integrates RTS/CTS and ACK to further enhance the reliability feature. All these design goals enable it to produce a high PDR at link level, as well as, at route level.

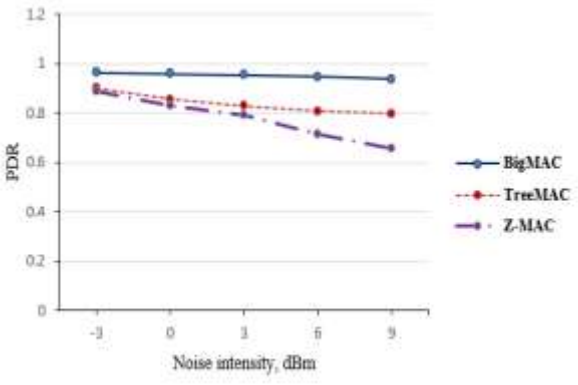

Fig. 6. PDR for different noise intensity level.

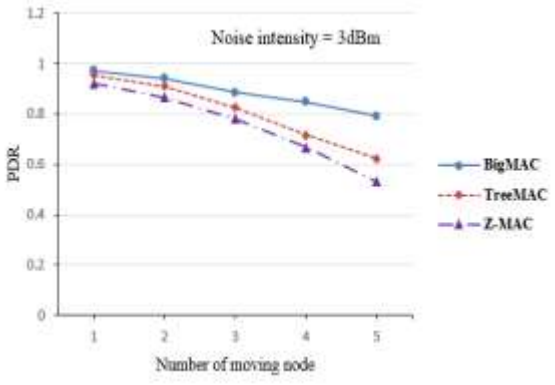

Fig. 7. PDR for the number of moving nodes.

In Fig. 6, PDR of BigMAC is decreasing with an increasing in noise intensity level. As the link break rate increases with an increase in noise intensity, the PDR of all protocols decreases. However, unlike other two protocols, BigMAC can improve PDR by using alternate parents and opportunistic slot reusing. The situation becomes worse when the effect of node movement is considered along with noise intensity. We take the median value of noise intensity *i.e.*, 3dBm in this case. The net effect is that PDR of all three protocols decrease more than before. Compared to BigMAC, the PDR of TreeMAC and Z-MAC is very poor due to the complex slot scheduling in frequently changing link quality as shown in Fig. 7.

- Energy consumption

WSNs need to reduce the energy consumption to prolong their lifetime as battery replacement may not be often trivial. A high depth sensor node usually covers a remote area, which is generally unattended. Therefore, the energy saving of a remote sensor node is more important than that of a nearby sensor node.

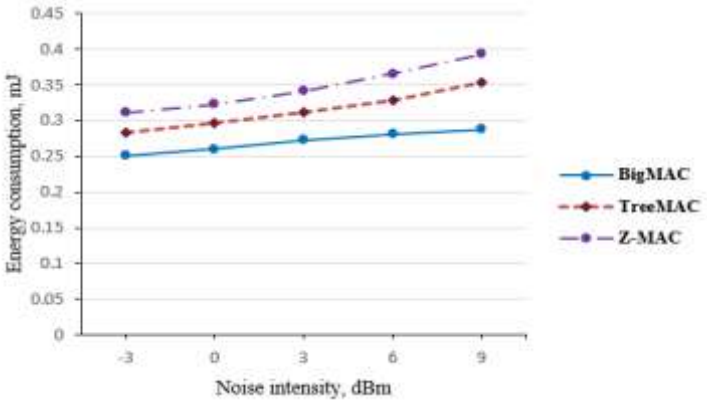

Fig. 8. Energy consumptions for noise intensity level.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

In spite of using relatively large slot, BigMAC can reduce the energy consumption significantly due to its aggregation and filtering, less frequent slot scheduling, and data packet recovery. Without any local link recovery, TreeMAC and Z-MAC incur a high amount of energy in industrial applications as shown in Fig. 8.

- *Control overhead*

In this paper, the delay is assumed to be end-to-end delay *i.e.*, the delay incurs by a data packet from any sensor node to a sink node. A delay time is acceptable until it does not affect the real-time constraint of the intended applications.

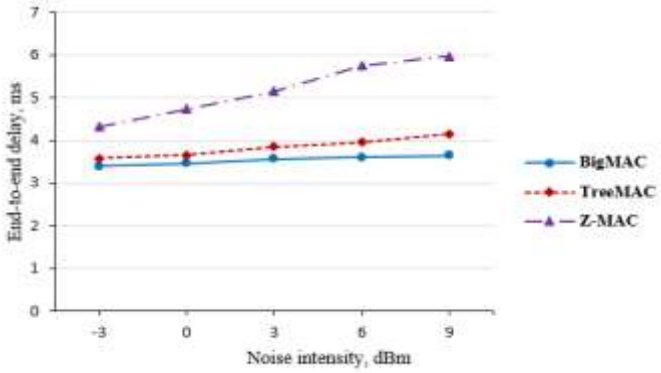

Fig. 9. Delay time for noise intensity level.

Since both BigMAC and TreeMAC consider the data propagation pattern of WSNs into their data transmission schedule, their data aggregation are very effective and thus their delay time seems to be reasonable as shown in Fig. 9. Between them, the weak link recovery method of TreeMAC makes a slight increase in delay compared to BigMAC. However, Z-MAC having its complex slot scheduling and less data aggregation capability gives a high delay, which may not be acceptable by the majority of industrial WSNs applications.

CONCLUSIONS

BigMAC employs a reliable tree construction as well as a reliable data transmission to encounter a packet loss. It also provides a run-time defense against an unpredictable link failure by delivering a data packet through an alternate parent. Moreover, a locally controlled CSMA operation within a relatively large time slot makes BigMAC a robust MAC protocol for any dynamic WSNs applications.

The simulation result shows that BigMAC clearly outperforms TreeMAC and Z-MAC every performance index. BigMAC is able to retain its performance at a satisfactory level when we impose a complex type of interference. Therefore, BigMAC should be a promising MAC protocol for a noisy and hazardous industrial applications. Our future work is to implement BigMAC in a real testbed to justify our simulated results.

REFERENCES

[1] H. Oh and P. V. Vinh, “Design and Implementation of a MAC Protocol for Timely and Reliable Delivery of Command and Data in Dynamic Wireless Sensor Networks,” Sensors, vol. 13(10), pp. 13228-13257, 2013.
[2] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, “Z-MAC: a hybrid MAC for wireless sensor networks,” IEEE/ACM Transactions on Networking, vol. 16(3), pp. 511-524, 2008.
[3] I. Rhee, A. Warrier, J. Min, and L. Xu, “DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks,” in Proceedings of the 7th ACM international symposium on mobile ad hoc networking and computing, Florence, Italy, pp.190-201, 22-25 May 2006.
[4] W.–Z. Song, R. Huang, B. Shirazi, and R. LaHusen, “TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks,” in Proceedings of IEEE International Conference on Pervasive Computing and Communications, TX, USA, pp. 1-10, 9-13 March 2009.
[5] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, “Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks,” in Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, Colorado, USA, pp. 293-306, 31 October - 3 November 2006.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

[6] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The Format War Hits the Factory Floor," IEEE Industrial Electronics Magazine, vol. 5(4), pp. 23-34, 2011.
[7] IEEE Standard for Local and metropolitan area networks - Part 15.4: (2011). Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std. 802.15.4-2011. Available online: http://standards.ieee.org /findstds/standard /802.15.4-2011.html.
[8] P. Suriyachai, J. Brown, and U. Roedig, "Time-Critical Data Delivery in Wireless Sensor Networks," in Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems, Santa Barbara, CA, USA, pp. 216-229, 21-23 June 2010.
[9] M. X. Cheng, X. Gong, Y. Xu, and L. Cai, "Link Activity Scheduling for Minimum End-to-End Latency in Multihop Wireless Sensor Network," in Proceedings of IEEE Global Telecommunications Conference, Houston, TX, USA, pp. 1-5, 5-9 December 2011.
[10] H. Le, J. V. Eck, and M. Takizawa, "An Efficient Hybrid Medium Access Control Technique for Digital Ecosystems," IEEE Transactions on Industrial Electronics, vol. 60(3), pp. 1070-1076, 2013.
[11] J. Silvo, L. M. Eriksson, M. Bjorkbom, and S. Nethi, "Ultra-reliable and Real-time Communication in Local Wireless Applications," in Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, pp. 5611-5616, 10-13 November 2013.
[12] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," Wireless Communications & Mobile Computing, vol. 7(7), pp. 863-875, 2007.
[13] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, "Aggregation convergecast scheduling in wireless sensor networks," Wirel. Netw., vol. 17, pp. 319-335, 2011.
[14] L. Chang, B. Zhang, L. Cui, Q. Li, and Z. Miao, "Tree-based delay guaranteed and energy efficient MAC protocol for wireless sensor network," in Proceedings of International Conference on Industrial Control and Electronics Engineering, Xi'an, China, pp. 893-897, 23-15 August 2012.
[15] M. Arifuzzaman, M. Matsunoto, and T. Sato, "An Intelligent Hybrid MAC with Traffic-Differentiation-Based QoS for Wireless Sensor Networks," IEEE Sensor Journal, vol. 13(6), pp. 2391-2399, 2013.
[16] T. D. Han and H. Oh, "Quasi-tree mobility for internet connectivity of mobile ad hoc network," Wirel. Netw., vol. 17, pp. 493-506, 2011.
[17] C. T. Ngo, and H. Oh, "A tree-based mobility management using message aggregation based on a skewed wait time assignment in infrastructure based MANETs," Wirel. Netw., vol. 20, pp. 537-552, 2014.
[18] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, pp. 95–107, 3–5 November 2004.
[19] E. Wandeler and L. Thiele, "Optimal TDMA Time Slot and Cycle Length Allocation for Hard Real-Time Systems," in Proceedings of the 11th Asia and South Pacific Design Automation Conference, Yokohama, Japan, pp. 479-484, 24-27 January 2006.
[20] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," in Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, Boston, MA, USA, pp. 125–138, 6–9 June 2004.
[21] C. A. Boano, X. Zu, M. A. Figa, T. Voigt, A. Willig, and K. Romer, "The Triangle Metric: Fast Link Quality Estimation for Mobile Wireless Sensor Networks," in Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN), Zurich, Switzerland, pp. 1–7, 2–5 August 2010.
[22] C. –Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, pp. 266–279, 5–7 November 2003.
[23] L. T. Dung and B. An, "A modeling framework for supporting and evaluating performance of multi-hop paths in mobile Ad-Hoc wireless networks," Comput. Math. Appl., vol. 64, pp. 1197–1205, 2012.
[24] C. Bettstetter, H. Hartenstein, and X. Perez-Costa, "Stochastic properties of the random waypoint mobility model," Wirel. Netw., vol. 10(5), pp. 555-567, 2004.