https://dx.doi.org/10.22161/ijaers.4.3.38 *ISSN: 2349-6495(P) | 2456-1908(O)*

A Slotted-sense Streaming MAC for Real-time Multimedia Data Transmission in Industrial Wireless Sensor Networks

Md Abul Kalam Azad[1], Amina Khatun[1], Md Abdur Rahman[2]

[1]Department of Computer Science & Engineering, Jahangirnagar University, Dhaka, Bangladesh
[2]Department of Mathematics, Jahangirnagar University, Dhaka, Bangladesh

***Abstract*—*To attain a flawless real-time data transfer over a communication channel of Industrial Wireless Sensor Networks (IWSNs) is a difficult issue. The reason is that data transmission reliability degrades remarkably due to the dynamically changing network topology. Although TDMA-based protocols are widely used for real-time data transmission in IWSNs, they are not suitable for dynamically changing network topology. Therefore, we analyze the communication behavior of real-time multimedia data transmission and try to find out some performance enhancing factors. Then, we propose a slotted-sense scheduling scheme in which a big time slot is shared among the nodes at same tree-level. A sharable slot not only assures the time constraint but also improves the reliability of multimedia data greatly. Moreover, the block data transmission technique with an adaptive acknowledgement is proposed to optimize the size of a sharable slot. Finally, the experimental results show that our approach outperforms other standard MAC protocols for real-time multimedia data transmission in IWSNs.*

Keywords*— *Industrial, NACK, Re-transmit, SS-MAC, Streaming.

I. INTRODUCTION

Instead of paying the required attention to the underlying Medium Access Control (MAC) protocols, most of the contemporary research works tried to improve the routing protocols for increasing the data transmission efficiency of Wireless Sensor Networks (WSNs) [1]. In practice, a poorly designed MAC protocol often creates a bottleneck problem in data packet transmission, and hence a network sometimes does not work efficiently as expected. Such a shortcoming seriously affects the transmission of real-time multimedia data, which requires more channel bandwidth, more energy and incurs a higher delay than that of scalar data transmission [2].

Interference is a common phenomenon in industry. Therefore, WSNs intended to industrial sectors should address the interference from industrial applications. Due to its real-time nature of data acquisition, Time Division Multiple Access (TDMA) is frequently used in industry. However, TDMA-based protocols have to encounter at least two complex issues for industrial applications. *Firstly*, they require a frequent time synchronization among the deployed nodes in WSNs. We know that the frequency of time synchronization depends on the skew rate and slew rate of a local clock, as well as, the size of a time slot. For a fixed skew rate and slew rate, a protocol along with smaller time slot requires frequent time synchronization than that of larger time slot. *Secondly*, a particular node requires a hard and fast rule for allocating the precise number of time slots. Moreover, introducing a new node to a WSN or removing an existing node from a WSN requires a fresh slot allocation for the whole network, which seriously affects data transmission efficiency [3]. In industry, these weaknesses of TDMA-based MAC protocols adversly affect the data transmission efficiency of WSNs.

To address the above weaknesses, some of the contemporary Carrier Sense Multiple Access (CSMA)-based MACs, IEEE-802.11 [4], SMAC [5], and their variants DMAC [6], TMAC [7], RMAC [8], and MMSPEED [9] attempt to increase the data transmission efficiency and to reduce the time delay of multimedia data packets in WSNs. However, these MAC protocols cannot provide any specific tool for assuring the time constraint of data packets. Moreover, some of them do not consider the data transmission reliability, while others do not consider the time constraint of data transmission. MMSPEED [9] considers both aspects by allowing multipath data transmission. However, its multipath data transmission significantly increases the collisions among data packets, and increases the energy consumption as well for industrial WSNs.

The Distributed Coordination Function (DCF) of IEEE 802.11 standard is modified for enhancing the real-time data transmission in wireless networks [10-12]. The findings of [10] are that the throughput increases with an increase in the number of backoff stages, and that the average number of transmission per packet decreases with an increase in the size of the initial backoff window. The works [11-12], based on the contention design of [10], propose to optimize the retry limit for reducing the delay of multimedia data transmission using IEEE 802.11 DCF. However, these works are intended for AdHoc wireless networks where the fairness requires a different type of resource allocations among users than that of WSNs.

In this paper, we propose a reliable MAC protocol for multimedia data streaming termed as Slotted-sense Streaming MAC (SS-MAC), which exploits the transmission characteristics of multimedia data in WSNs. Instead of using either CSMA or TDMA-based slot scheduling, we use a hybrid slot scheduling in which a big slot is shared among the nodes at same tree level so that the nodes can share the big slot using CSMA. The time constraint is preserved by the size of big slots whereas the robustness of data delivery is achieved by switching to alternate parents through the contention process within a big slot. At any time, only the nodes of two adjacent tree levels are active in which the nodes at high tree level are the sender and the nodes at low tree level are the receiver of multimedia data packets. If a node with multimedia data wins over a channel, it sends all fragmented packets to a direct parent node or an alternate parent node in a reliable and timely manner. When a node finishes data packet transmission, it will be informed with a Negative ACKnowledgement (NACK) message about the corrupted or lost data packets if there is any. Consequently, the sender re-transmits the corrupted data packets in a burst for the maximum allowed number of times, which is adaptively determined. Therefore, SS-MAC is able to overcome the difficulties of dynamically changing wireless channel, and hence outperforms other MAC protocols in data transmission, as well as, energy consumption efficiency.

The paper is organized as follows: Section 2 deals with some related works. Section 3 describes the motivation and problem statements underpinning the expectations from SS-MAC. The step by step design of the protocol is discussed in section 4. Section 5 discusses some experimental results followed by some concluding remarks in section 6.

II. RELATED WORKS

In IEEE 802.11 [4], a familiar communication protocol for wireless networks, a sender exchanges Request-to-Send (RTS) and Clear-to-Send (CTS) control messages with a receiver to reserve a communication channel only for the first fragment of a data packet and the corresponding ACKnowledgement (ACK). The first fragment and its ACK reserve the channel for the second fragment of a data packet and its ACK, and so on. However, when the sender transmits a fragment of a data packet, it releases the channel and monitors the channel for an ACK of previously transmitted data fragment. Relinquishing the medium in such a way requires an extra contention time and hence, a data packet suffers from high delay. If an ACK of any data fragment is lost, the sender must leave the transmission process in an incomplete stage, and re-contend for the channel to transmit the remaining fragments.

A duty cycle version of IEEE 802.11, SMAC [5] fragments a long message into many small pieces in order to transmit them in burst. Only one pair of RTS and CTS control packet reserves the channel for transmitting all of the data fragments. The sender waits for an ACK for each transmitted fragment. In case of a failure, it extends the reserved transmission time for this fragment, and then retransmits the current fragment immediately. The logic behind of sending an ACK for each fragment is to avoid the hidden terminal problem at receiver premises. This causes an extra delay for a large data packet in real-time multimedia transmission. As the integrity of a data packet must be maintained, the successful reception of an isolated data fragment is meaningless to a multimedia data application. Therefore, an ACK for each individual data fragment wastes the precious time and bandwidth of a communication channel.

In DMAC [6], a staggered wakeup schedule is designed for streaming of data packets from the sources to a sink. Instead of implementing the RTS and CTS handshaking for reducing the control overhead, the protocol keeps the basic active cycle with a Receive + ACK period followed by a Transmit + ACK period in a data transmission slot. If an ACK is not found, the corresponding data packet is sent in the next active transmission slot within the maximum of three retransmission attempts. For transmitting multiple data packets, the nodes on the multi-hop path need to increase the duty cycle adaptively. The protocol does not propose the streaming of multimedia data fragments and does not provide the influence of data streaming on network performance.

On the other hand, each node in TMAC [7] transmits its queued packets in a burst at the start of a frame. This adaptive duty cycle MAC determines some activation events, and then an inactive period after which a node can go to sleep for the remaining cycle time. The protocol reduces idle listening by transmitting all messages in bursts of variable length, and sleeping between two bursts. Though the protocol effectively addresses the data forwarding pattern of small packets, it fails to consider the transmission of large multimedia data packets in which the fragmentation is mandatory for channel efficiency. Moreover, the assumed error free channel for data transmission is very unrealistic in any industrial WSNs applications.

Toward designing a reliable MAC, RMAC [8] adopts both the explicit and the implicit ACK for reliability assurance. Every intermediate node skips backoff period and immediately transmits a successfully received packet to the next forwarding node. By overhearing the data transmission of next forwarding node, each node considers its transmission as successful, which is treated as an implicit ACK. As there requires no transmission beyond a sink, an explicit ACK is required for the immediate neighbors (1-hop away) of a sink. The process indeed reduces the delay time of data streaming; however, if the next hop node requires an extended amount of time to get the channel for data transmission, the implicit technique incurs even more delay than other techniques. Moreover, the protocol suffers from the poorly designed retransmission attempts, which allows only one increment in retransmission attempts for any types of error from a communication channel.

MMSPEED [9], a cross layer approach, adopts the reliability and real-time Quality-of-Service (QoS) over multipath data transmission. The protocol differentiates various services by mapping the data packets to different priority queues. That it sends multiple copies of same data packet over multiple paths enhances the real-time constraints and reliability of data transmission. However, this way of packet delivery gives high packet collisions and consequently, it incurs more retransmission attempts and delay in industrial WSNs. The protocol does not consider the energy consumption efficiency and fragmentation of data packets. Moreover, a node of this protocol requires multicasting support as well as a large amount of memory to remember multiple paths. Therefore, this protocol is not suitable for the real-time multimedia data transmission in WSNs.

Last but not least, it is worth mentioning to discuss about *SMAC with Block Acknowledgement* [13] – we termed it as BA-MAC - an important work on the fragmentation of data packets. It introduces the Block Acknowledgement (BA) concept from IEEE 802-11e standard [14] in SMAC. The protocol improves the packet delivery efficiency and reduces end-to-end delay by fragmenting a large data packet. However, it is not free from data forwarding interruption problem as off SMAC. Additionally, some key deficiencies such as formation of variable logical groups, duplicate packet discarding and impractical arrival time of packets *etc*., are observed in the protocol operation.

III. BACKGROUND

3.1 Motivation and Problem Statements

In industry, the vibration rate of the generated signal by various appliances is in the range of 40 KHz [15]. If we want to build a safety monitoring system for appliances, the sensed signals from sensor nodes are required to be sent to a sink. The received signals are required to be analyzed to know whether a safety measure is required or not. To get a functional system, the sensed signal is required to follow the "*Nyquist-Shannon*" sampling rate [16], which states that the sampling frequency should be at least twice of the highest frequency of sensed signal (input signal) to reproduce the monitoring phenomenon at a sink node as follow:

$$F_s \geq 2B \tag{1}$$

In (1), F_s stands for the sampling frequency of sensed signal and B stands for the highest frequency of the sensed signals at a sensor node. Therefore, the minimum required bandwidth (data rate) by a node should be 320 Kbyte/sec, where 4 byte single precision FPA-IEEE 754 [17] coding is used for floating point integer representation as follow:

$$\begin{aligned} BW_{node} &= 80\,KHz \times 4\,Bytes \\ &= 320\,KByte/sec \end{aligned} \tag{2}$$

The above analysis shows that the amount of data to be transmitted for a simple vibration signal requires a high channel bandwidth. Therefore, any multimedia data such as still images, audios and videos require much more bandwidth than that of scalar data. The bandwidth calculation in (2) is for a noise-free channel. In practice, a noise-free wireless channel in industrial applications does not exist. In a noisy channel, more retransmissions are taking place due to the corrupted or lost data packets. Therefore, a high volume of data is required to be transmitted through an industrial WSN for such applications. Compression may be an effective tool for reducing the volume of transmitted data.

A 10:1 compression level results in a little distortion in transmitted multimedia data. This level of compression is able to reduce the data rate of WSNs to 30 packets/sec in which each packet contains of 8738 bits. However, transmission of such a large data packet through a noisy industrial channel is very inefficient. For the channel efficiency and minimum delay, small data packets are required to be transmitted through a noisy channel. Therefore, a data fragmentation technique is required so that a packet loss reduces the channel occupation. In this paper, we propose a MAC protocol that is able to transmit a large data packet reliably in a fragmented form at hop-by-hop basis instead of end-to-end basis [18] without any loss of simplicity.

3.2 Network Model

A WSN consists of one data collection and control server (referred to as simply *node* or *sink node*), and a number of sensor devices (referred to as simply *node* or *sensor node*) that includes at least one sensor module for thermal, gas, oxygen, smoke, or flame sensing. Each sensor node generates a multimedia data packet from sensed data, fragments them accordingly, and sends them to a sink node within a specified time bound defined by an industrial application.

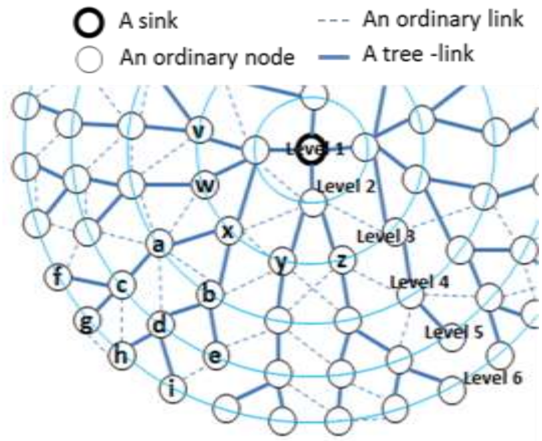

Fig. 1: A network model

Sensor nodes form a tree originating from a sink node, and each sensor node has a parent and may have multiple children. A node is said to be a *tree-node* if it belongs to a tree. Otherwise, it is an *orphan-node*. Two nodes that can directly and mutually communicate with each other are said to have a *link* or an *ordinary link*. Especially, a link between a node and its parent is called a *tree-link*.

Fig. 1 shows a network model of a tree originating from a sink node where the tree has a tree level (depth) of six. The solid lines and the dashed lines indicate *tree-links* and *links*, respectively.

3.3 Notations and Definitions

We use some notations as follows:

TABLE I: Notations

- *level*(*i*): The level of node *i*
- *N*(*i*): A set of neighbors of node *i*
- *N*-1(*i*), *N*=(*i*), *N*+1(*i*),: A set of neighbors of node *i* at *i*'s level-1, *i*'s level, *i*'s *level*+1, respectively. Then, $N(i) = N\text{-}1(i) \cup N{=}(i) \cup N{+}1(i)$
- *C*(*i*): A set of children of node *i*
- *D*(*i*): A set of descendants of node *i*
- $D[i] = D(i) \cup \{i\}$
- *P*(*i*): The parent of node *i*
- *CN*(*i*): A set of the nodes with which a node *i* competes to acquire a channel.
- $CN[i]$: $CN(i) \cup \{i\}$
- *H*: The depth of a tree

For the convenience of description, we define some terminologies as follows:

A link (*a*, *b*) is defined to be *bidirectionally reliable* (*B*-reliable) iff a directional link from *a* to *b*, and *a* directional link from *b* to a are both reliable. If a tree is constructed along with *B*-reliable links, the tree is said to be a *B-reliable tree*.

A *sharable slot* (SS) is a time span in which all nodes at the same tree level share to receive data packets from their children and transmit their data packets to their respective parents using CSMA. SS(*i*) is used to denote a sharable slot allocated to level *i* and divided into two parts, $SS^{Rx}(i)$ and $SS^{Tx}(i)$ where the nodes at level *i* use to receive data packets from their children and to transmit data packets to their parents, respectively.

Definition 1: A superframe (*SF*) is given by the sum of the transmission portions of all the sharable slots allocated to different levels of a tree as follows:

$$SF = \sum_{i=2}^{H} SS^{Tx}(i) \qquad (3)$$

where $SS^{Tx}(i)$ and $SS^{Tx}(j)$ when slot *i* and slot *j* do not overlap.

An application states the *SF* as the maximum time that a sink can wait to acquire data packets from all participating nodes. For any loss of generality, we use big-slot as a slot when it is used in our SS-MAC protocol.

IV. FORMULATION OF PROPOSED PROTOCOL

Most of the MAC protocols show some limitations for their apathetic role in categorizing the impaired received packets. Impairment in the received packet can be occurred for a variety of reasons such as noise burst, collisions by multiple senders at the receiver premises, collision between different destined packets for the overlapped collision domains. Moreover, packet may be lost altogether due to multipath fading and congestion at a choke point in the forwarding path of data packet transmission. Most of the protocols generally address all of these problems by packet re-transmission *i.e.*, re-sending the lost or corrupted packet with doubling the size of a contention window [4].

Doubling the contention window has a serious consequences on the time delay and hence, on the data delivery efficiency of WSNs. In case of congestion and hidden terminal problems, increasing contention window allows the competing nodes an additional time span to choose its mutually exclusive slot to avoid collisions. However, the two common problems in successful data packet reception: *data collision* and *data loss*. These should be handled differently. Using the same solutions for both will produce unnecessary delay and thereby, waste costly bandwidth of wireless channel. We, therefore, need to analyze the topological structure and operational behavior of WSNs to design an effective data transmission protocol for multimedia applications.

4.1 Collision behavior of a sensor network

Collision that seems to be a usual phenomenon in WSNs should be carefully addressed by any MAC protocols. In a contention based tree protocol, there are some specific locations of a tree in which the probability of collisions might be high compared to other regions. We all know that collisions are receiver oriented. Therefore, if the receiver nodes can be kept at mutually exclusive collision domains, then data packets are safe from collision. On the other hand, the intermediate node acting as a relay in the sensor network paradigm may expose to collision when either its parent or children simultaneously try to transmit data packets.

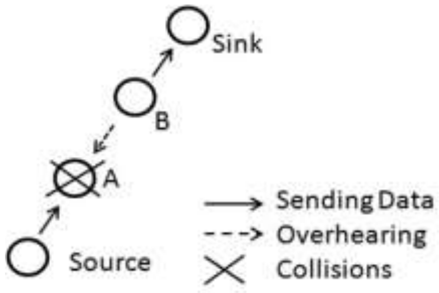

Fig. 2: Collision at node-A due to simultaneous neighbor transmissions

In Fig. 2, node B is transmitting to the Sink, and node A is overhearing the transmission phenomenon. At the same time, if the a source unaware of the transmission of B tries to transmit, a collision happens at node A. Multiple children can also make a collision to their receiving parent node at the branching point of a subtree as shown in Fig. 3.

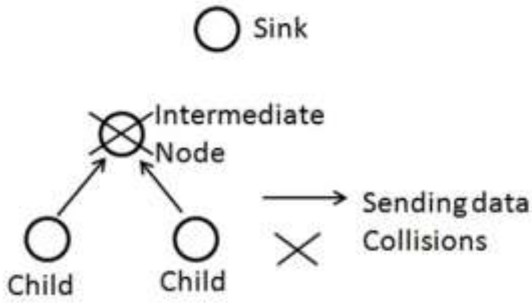

Fig. 3: Collision for simultaneous activation from the data transmission of children

Nodes in the parallel paths within the overlapping collision domain of each other may interfere when their data sending scheduling time coinsides as in Fig. 4. It is interesting to note that the collision probability is more severe near the sink node rather than source nodes as all of the routes converge to a common sink.

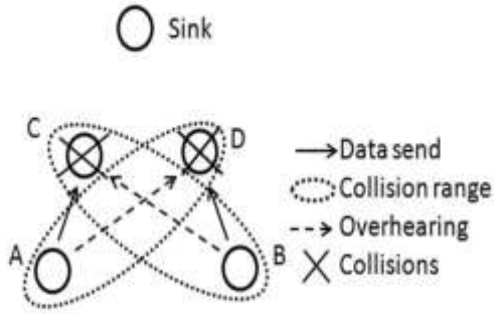

Fig. 4: Collisions for the overlapping interference domain

4.2 Overhearing behavior of sensor network

All of the immediate neighbors of both the sender and receiver should keep silent after they hear the RTS and CTS packet for the first time until the current transmission is over [5]. Sketching the process in Fig. 5, it can be concluded that within five successive nodes along the path, only one node can transmit its data to its neighbor toward the sink. The remaining three nodes can go to sleep to conserve energy by setting their appropriate wake up timer. However, the farthest node from sink in this group has some flexibility *i.e.*, it can receive packet from its children if necessary. So, the immediate neighbor nodes of sender and receiver should not take part in any transmitting and receiving actions.

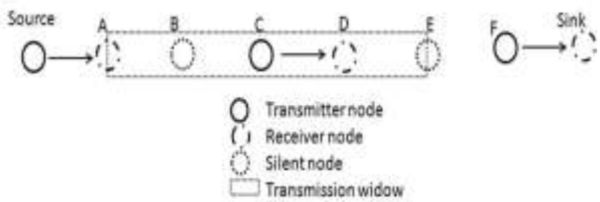

Fig. 5: Showing 4-Hop blocking by a transmitting node

In Fig. 5, node C is transmitting to node D while other nodes adjacent to them along the path have nothing to do. So, node B and node F can easily go to sleep for the time duration of a packet transmission. But node A can receive packets if the source has any for reducing the packet delay of the overall transmission process.

4.3 Determining ACK and re-transmission attempts

In order to guaranttee the reception of all fragments at a receiver, the appropriate type of ACK is necessary for the real time multimedia applications. As the single fragment of a multimedia packet is meaningless, they all need to be transmitted to the next forwarding node along the path almost simultaneously. Thus introducing an ACK at every fragment produces high transmission delay, which may be the cause of failing the mission altogether. As the data packet transmission in a WSN is generally unidirectional - from all data sources to a sink – the modified Negative ACK (NACK) is more plausible than positive ACK. In

https://dx.doi.org/10.22161/ijaers.4.3.38 *ISSN: 2349-6495(P) | 2456-1908(O)*

other words, when a data fragment of a burst is corrupted, the receiver will notify the sender about the corrupted fragment after the completion of current burst.

To adaptively combating the volatile behavior of a lossy channel, the Bite Error Rate (BER), e_{ber} of a transmission channel is directly calculated from the Signal to Noise Ratio (SNR) sensed at a receiver. At the end of the burst, the receiver informs the sender about the fragment loss, and retransmission attempts (*ReTX*). *ReTX* is calculated as in (4) at the receiver end and piggybacked on the NACK.

$$ReTX = ReTX_{max} - \left\lceil \log_{10}\left(\frac{1}{e_{ber}}\right) \right\rceil \quad (4)$$

$$where, ReTX_{max} \text{ is the threshold}$$

The second term in the right side of (4) decreases with an increase of the BER value of the corresponding channel and therefore, the number of adaptive re-transmission attempts, *ReTX* will increase, and *vice versa*. $ReTX_{max}$ may be a tuning parameter depending on the characteristics of a channel. Thus an initial $ReTX_{max}$ value can be set empirically and then, optimized through experiment. After taking the maximum number of attempts, the packet is considered to be undeliverable and discarded from the system.

4.4 Data Forwarding in SS-MAC

From the above analysis, we conclude that in a carefully planned MAC protocol for multimedia applications, per hop data delivery plays a crucial role in the overall success of a system. Influenced by the idea of message passing [5] designed for handling unorthodox size packet in scalar sensor network, we propose SS-MAC protocol for multimedia applications with an adaptive re-transmission technique.

In our work, the large multimedia packet is broken down into several small sized fragments, and transmitted them in burst as in Fig. 6. The channel is allocated by *RTS+CTS* contention method only at the first slot of data transmission. Afterwards, sender will stream the remaining fragments in a nonintermittent way.

Input: multimedia packet (*P*) and fragment size (*s*)

1. Start of sending process
2. Divide *P* of size *s* into *n* fragments
3. Augment *n* and *ID* in each fragment
4. $for \leftarrow 1\ to\ RTS_{max}$
   a. Send *RTS*
   b. If it hears nothing, *continue*
   c. If it hears a *CTS* packet
      i. Stream all the fragmented packets with necessary interval; and *break*

   end for
5. *If* it gets a *NACK* within a timeout interval after last packet
   a. Retransmit the fragments specified by *NACK*

   else

   b. Change state accordingly

   end if
6. End of sending process

Fig. 6: Block sending process within a big-slot

After sending the final fragment, a receiver will acknowledge the sender about the failure of any fragments as in Fig. 7. For successful transmission, sender will wait for an ACK and start sending the next multimedia packet when ACK timer espires. In case of failure, sender will retransmit the missing data fragments for a number of times specified by the receiver depending on the condition of a channel. The following algorithm clearly states the process.

Input: Fragments (*f*), and total number (*n*) of fragments

1. Start of receiving process
2. *If* it senses a *RTS* packet
   a. Send a *CTS* packet to the sender
   b. Wait for the timeout interval
   c. If nothing happens, *Exit*

   end if
3. Buffer all transmitted data fragments
4. Inspect the checksum
5. *If* *n* fragments have received correctly
   a. Reconfigure this node as the future sender of the packet
   b. Change state accordingly, and Exit

   else

   c. Construct an *NACK* packet with *ID* of corrupted fragments and *ReTX*, and do following:
      i. Send the NACK packet to the sender
      ii. Receive the retransmitted copies
      iii. Fuse the corrected copies in order
      iv. Reconfigure this node as

https://dx.doi.org/10.22161/ijaers.4.3.38 *ISSN: 2349-6495(P) | 2456-1908(O)*

the future sender of the packet
V. Go to the sending state, and Exit
end if
6. End of receiving process

Fig. 7: Block receiving process within a big-slot

V. SIMULATION EXPERIMENTS

5.1 Simulation Environment

In order to evaluate the performance of SS-MAC, a set of simulation experiments have been performed using NS-2.34 simulator using random waypoint model. We compare our proposed SS-MAC with IEEE 802.11-based MAC, SMAC and BA-MAC for various simulation scenarios. The simulation parameters and values are given in Table II. The simulation for each scenario was performed five times, and then the average value for each metric was calculated.

TABLE II: Simulation parameters

Parameter	Value
Simulator	NS-2.34
Mobility model	Random waypoint
Traffic	CBR
Transmission range	20m
Number of nodes (*MAX*)	50
Terrain size (*a* x *a*)	*a* = 100m
Maximum speeds	5, 10, 15 m/s
Pause time	30s
Number of sessions	15
Simulation time	600 seconds
Packet size	512 bytes
Packet transmission rate	4 packets/s
Channel Bit Rate	1 Mbit/s
Slot time, *T*	20 µs
Retransmit Limit, *ReTX*	2 ~ 4 (Adaptive)
Buffer Size	40 Packets
BER	$10^{-2} \sim 10^{-4}$
Transmit Power	28 mW

To compare SS-MAC with other MAC protocols, we use some metrics such as energy consumption, packet delivery ratio and end-to-end delay for multihop WSNs.

***Packet delivery ratio* (*PDR*):** It indicates the ratio of the total number of data packets received at destinations to the total number of data packets generated during simulation.

$$PDR = \frac{\sum_{i \in S} nPacketsReceived\,(i.d)}{\sum_{i \in S} nPacketsSent\,(i.s)} \quad (5)$$

where S is a set of sessions created during the simulation, and *nPacketsReceived(i.d)* and *nPacketsSent(i.s)* are the number of packets received at a destination *d* and sent from a source *s* for the session *i*, respectively.

***End-to-End Delay* (*E2ED*):** The average time taken to deliver a data packet from its source to its destination is as follows:

$$E2ED = \frac{1}{R} \sum_{i=1\ldots n} \sum_{j=1\ldots k_i} \left(r(p_{ij}) - t(p_{ij}) \right) \quad (6)$$

where *R* is the total number of packets received by all destinations, p_{ij} is the sequence of j^{th} packets received at node *i*, and $t(p_{ij})$ and $t(p_{ij})$ is the receiving and transmitting time of p_{ij}, respectively.

According to [20], the CC2420 radio consumes 23 mA current in a receiving or listening mode, 8.5 mA current when transmitting at −25 dBm, 21 µA current in an idle mode, and 1 µA current in a sleep mode. Therefore, due to measure the energy consumption of a sensor mote, we count the amount of time that each node has to spend in a particular mode: *sleep*, *idle*, *receiving* or *transmitting*. Then, energy consumption of a node is calculated by multiplying the cumulative time stayed at each mode and power consumed to operate the radio in that mode, considering a battery of 3 volts. In this way, energy consumption is measured indirectly because the direct measurment of current imposes an additional burden on the low-powered sensor motes.

5.2 Simulation Results

To maximize data transmission efficiency over an error prone wireless channel, the optimum size of a data packet should be around of 152 byte for any terrestrial communication [19]. Since the size of a multimedia packet is large, we divided a multimedia data packet into a number of data fragments (usually 125 byte), and add the necessary PHY and MAC headers for increasing transmission efficiency.

The simulation results of our propose SS-MAC along with the other three standard MAC protocols are shown in in this section. In Fig. 8, packet delivery ratio (PDR) is simulated of a sensor node at different levels (depths) in a tree network. It is shown that the PDR for SS-MAC is higher than other three MACs, especially far better than 802.11-based MAC and SMAC protocols. The reason for lower PDR in 802.11-based MAC and SMAC is that they are intended for small scale data packet, as well as, for single hop data transmission. When it comes to the question of multimedia and multihop data transmission, those two protocols show lower PDR than others due to their less adaptivable nature toward the changing of channel conditions.

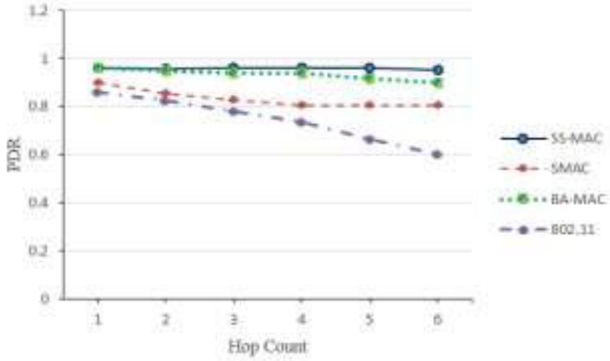

Fig. 8: Packet delivery ratio at different depths of a tree

BA-MAC, on the other hand, shows a better PDR than those two protocols. However, its packet-wise ACK makes its PDR lower than the proposed blockwise ACK-based SS-MAC protocol.

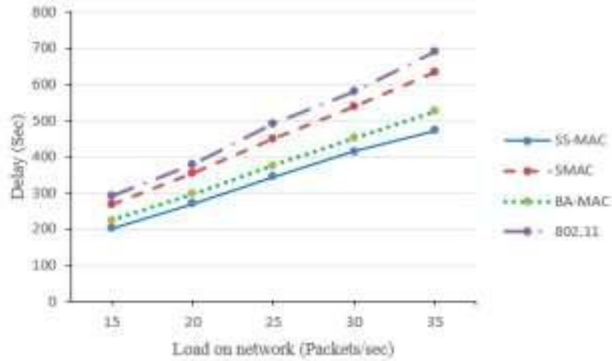

Fig. 9: End-to-end delay for various load levels

Fig. 9 depicts the end-to-end delay for data packets when the network load is varied. As of the figure, SMAC and 802.11 rise more sharply than SS-MAC, and BA-MAC rises marginally compared to SS-MAC. Though the delay reduction technique used in our proposed MAC seems to be insignificant compared to BA-MAC for single hop data transmission, this small delay reduction gives a significant improvement in multihop data transmission. The interesting outcome is that both 802.11 and SMAC fail to achieve the real-time constraints quickly as the load increases. However, our proposed MAC is less sensitive to network load and can easily support multimedia data transmission in WSNs.

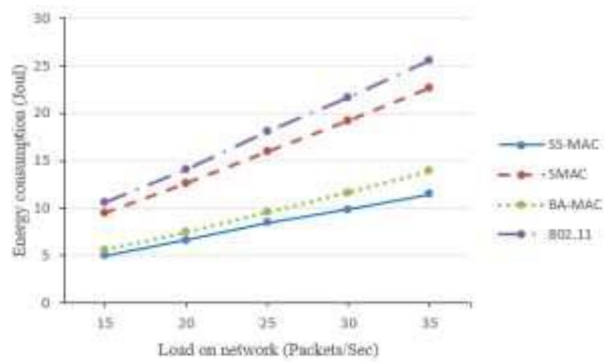

Fig. 10: Energy consumption under different loads

Due to space constraint, we do not provide the detail calculations of energy consumption efficiency. Our proposed SS-MAC is inherently energy efficient for its dynamic slot allocation ability, data aggregation and filtering. Moreover, we are able to reduce a sufficient amount of energy consumption by reducing the control overhead. Fig. 10 shows that our proposed protocol outperforms the other two standard MAC protocols, while it marginally outperforms BA-MAC. In summary, we can say that the proposed SS-MAC shows the better performance than other three standard protocols in every aspects.

VI. CONCLUSION

In this paper, we have developed SS-MAC protocol capable of reducing the transmission delay for real-time multimedia data in industrial WSNs. For such type of applications, we discovered some suitable parameters through analyzing the contention characteristics of wireless nodes with a due consideration on existing MAC protocols. Moreover, our modified formula for adaptive retransmission of multimedia data packets gives an improved data transmission rate. In simulation results, SS-MAC shows the superior performance over the 802.11-based MAC, SMAC and BA-MAC in improving data transmission efficiency as well as in reducing delay and energy consumption. Therefore, we conclude that SS-MAC can be an efficient MAC protocol for the real-time multimedia data transmission in industrial WSNs.

REFERENCES

[1] L. Shi, A. Fapojuwo, N. Viberg, W. Hoople, and N. Chan, "The Effectiveness of QoS Constrained AODV Routing for Voice Support in Multi-Hop IEEE 802.11 Mobile Ad Hoc Networks," in IEEE Wireless Communications and Networking Conference (WCNC-2009), pp. 1-6, Hungary, 2009.

[2] I. F. Akyildiz, T. Melodia, and K.R. Chowdury, "Wireless Multimedia Networks: A survey," IEEE Wireless Communications, vol. 14, no. 6, pp. 32-39, 2007.

[3] P. Suriyachai, U. Roedig, and A. Scott, "A Survey of Protocols for Mission-Critical Applications in Wireless Sensor Networks," IEEE Communications Survey and Tutorials, vol.14, no. 2, pp. 240-264, 2012.

[4] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1999 edition, 1999.

[5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for

wireless sensor networks," IEEE/ACM Trans. Netw., vol. 12, pp. 493-506. June 2004.

[6] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks," in 18th Int. Parallel and Distributed Processing Symp., pp. 224-231, Santa Fe, NM, USA, 2004.

[7] T. V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in 1st ACM Conf. Embedded Networked Sensor Systems, pp. 171-180, Los Angeles, CA, USA, 2003.

[8] R. Biswas, V. Jain, C. Ghosh, and D. P. Agrawal, "On-Demand Reliable Medium Access in Sensor Networks," in 7th IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks, WoWMoM'06, pp. 251-257, Buffalo, NY, USA, 2006.

[9] E. Felemban, C. –G. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," IEEE Trans. Mobile Comput., vol. 5, pp.738-754, June 2006.

[10] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal on Selected Areas in Communications, Vol. 18, no. 3. March 2000.

[11] P. Chatzimisios, A. C. Boucouvalas, and V. Vitas, "IEEE 802.11 Packet Delay – A Finite Retry Limit Analysis," IEEE Global Telecommunications Conference, GLOBECOM'03, vol. 2, pp. 950-954, San Francisco, USA, 2003.

[12] B. J. Oh and C. W. Chen "Analysis of Retry Limit for Supporting VoIP in IEEE 802.11 EDCA WLANS," in 16th International Conference on Computer Communications and Networks, ICCCN'2007, pp. 464-469, Hawaii, USA, 2007.

[13] Y. –T. Park, P. Sthapit, D. -H. Lee, Y. –S. Choi, and J. –Y. Pyun, "Data Fragmentation Scheme with Block ACK in Wireless Sensor Networks," in 5th FTRA International Conference on Multimedia and Ubiquitous Engineering, pp. 79-83, Crete, Greece, June 2011.

[14] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Std 802.11e-2005 edition, 2005.

[15] J. Molina, J. M. Mora-Merchan, J. Barbancho, and C. Leon, "Multimedia Data Processing and Delivery in Wireless Sensor Networks," Wireless Sensor Networks: Application-Centric Design, Open Access, Chapter 23, December 2010.

[16] The Nyquist–Shannon Sampling Theorem, http:// http://ptolemy.eecs.berkeley.edu/eecs20/week13/nyquist Shannon.html

[17] H. Yokoo, "Overflow/underflow-free floating-point number representations with self-delimiting variable-length exponent field," in 10th Symposium on Computer Arithmetic, pp. 110-117, Grenoble, France, 1991.

[18] L. Parziale, D. T. Britt, C. Davis, J. Forrester, W. Liu, C. Matthews, and N. Rosselot, "TCP/IP Tutorial on Technical Overview," International Technical Support Organization, 8th Edition, IBM. December 2006.

[19] M. C. Vuran and I. F. Akyildiz, "Cross-layer Packet Size Optimization for Wireless Terrestrial, Underwater, and Underground Sensor Networks," in 27th Conference on Computer Communications, INFOCOM 2008, pp. 226-230, Phoenix, Arizona, USA, 2008.

[20] CC2420 Datasheet, TI. Available online: http://www.ti.com/lit/ds/swrs041b/swrs041b.pdf/ (accessed on 28 September 2013).