

# *A Fast, Adaptive, and Energy-efficient Multi-path-multi-channel Data Collection Protocol for Wireless Sensor Networks*

Cheng Kiat Tan  
Faculty of Information  
and Communication  
Technology  
Universiti Tunku Abdul  
Rahman  
Kampar, Perak, Malaysia  
zhjie12@utar.my

Soung-Yue Liew  
Faculty of Information  
and Communication  
Technology  
Universiti Tunku Abdul  
Rahman  
Kampar, Perak, Malaysia  
syliew@utar.edu.my

Hock Guan Goh  
Faculty of Information  
and Communication  
Technology  
Universiti Tunku Abdul  
Rahman  
Kampar, Perak, Malaysia  
gohhg@utar.edu.my

Ivan Andonovic  
Department of Electronic  
and Electrical Engineering  
University of Strathclyde  
Royal College Building  
204 George Street,  
Glasgow, G1 1XW  
United Kingdom.  
i.andonovic@strath.ac.uk

**Abstract**—Energy consumption, traffic adaptability, fast data collection, etc are the major issues in wireless sensor networks (WSNs). Most existing WSN protocols are able to handle one or two of the above issues with the other(s) being compromised. In order to reduce the energy consumption of wireless sensor nodes while having fast data collection under different traffic generating rates, this paper proposes a fast, adaptive, and energy-efficient multi-path-multi-channel (FAEM) data collection protocol. FAEM makes use of the Basketball Net Topology proposed in the literature, in which a multi-parent-multi-child connection table is pre-established at each node; each node is also pre-assigned a receiving channel which is different from those of the neighboring nodes so as to eliminate the transmission interference. During data transmission, time is divided into duty cycles, and each consists of two phases, namely distributed iterative scheduling phase and slot-based packet forwarding phase. The former is to match parents and children of the entire WSN in a distributed manner in order to determine whether a node should be in upload (to which parent), download (from which child), or sleep mode in a particular slot; while the latter is for nodes to take action according to the schedule. Simulation shows that our protocol is able to achieve lower energy consumption, data reliability and low latency even during a high traffic load.

**Keywords**—Wireless sensor network, energy efficiency, adaptive schedule, iterative matching, multi-channel

## I. INTRODUCTION

Wireless Sensor Network (WSN) is composed of sensor nodes that are deployed in a wide area to monitor environment conditions such as luminosity, temperature, humidity, etc [1]. A sensor node is a small device that consists of three basic components: a sensing subsystem for monitoring the environment condition, a processing

subsystem for data processing, and a wireless communication subsystem for data transmission.

Basically, the nodes can be divided into two categories which are sensor nodes and sink node(s). A sensor node is used to sense the environment condition(s), forward the collected data, and relay data for others to the sink node; while the sink node is the destination of the collected data from the sensor nodes. Typically, data are forwarded from sensor nodes to the sink through multi-hop communications. The data transmission and aggregation can be done through certain routing protocols or algorithms [2]. Each sensor node is required to have a power supply to perform the programmed task. This power source often consists of a battery with a limited energy budget. Therefore, prolonging the network lifetime has become a critical issue. The sensor node will die off when it runs out of its energy. This will affect the whole WSN because some nodes may be disconnected from the sink.

In order to prolong the lifetime and maintain the connectivity of WSN, different topologies have been proposed in the literature. For instance, a new topology called Basketball Net Topology (BNT) [3] has recently been designed for WSN to address the routing and resiliency issues. With reference to Figures 1 and 2, the BNT levelizes the nodes in the network and introduces a connection table to each node. The level of a node indicates the minimum number of hops through which a packet can be forwarded to the base station from the node, and thus a node may have multiple parent nodes, and each indicates a possible shortest path to the base station. On the other hand, each node will establish a connection table and this table can be divided into three rows, namely parent, peer and child rows, as shown in Figure 2. Comparing with cluster topology, each node will forward packets to its parent node

to ensure shortest path rather than indirectly going through the cluster head. In addition, this topology can be self-recovered if any one of the node fails to ensure the network reliability. Interested readers may refer to paper [3] for more details about the BNT. However, the energy-efficient data collection protocols have not been discussed in [3].

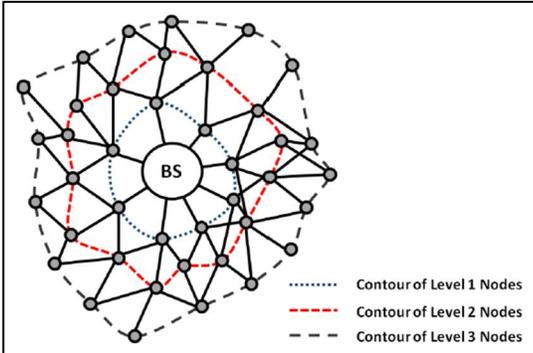


Fig. 1. Basketball Net Topology [3]

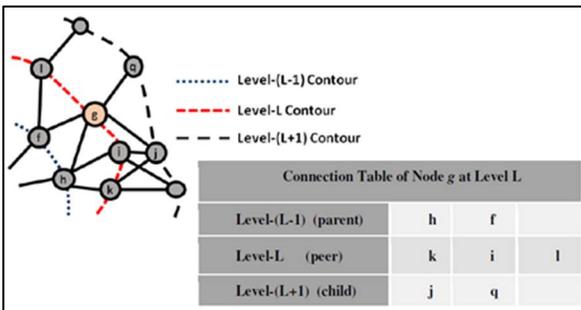


Fig. 2. Connection table of node g [3]

In this paper, we propose a fast, adaptive, and energy efficient multi-path-multi-channel (FAEM) data collection protocol based on the Basketball Net Topology. FAEM makes use of slot allocation with adaptive schedule based on traffic load. In FAEM, time is divided into duty cycles, and each cycle contains a scheduling phase and a packet forwarding phase. The multi-channel of data transmission is considered to prevent interference from each other node. Thus, this protocol is able to provide collision free during data transmission.

The rest of this paper is organized as follow: Section II presents the related work on energy saving for WSN. Section III describes the main design of our protocol. Section IV shows the simulation result of our protocol. Finally, the conclusion is drawn in Section V.

## II. RELATED WORK

Since the radio of node will consume a lot of energy, turning off the radio is one of the ways to save energy. However, the issue is how and when to inform a node to turn its radio on again when other nodes intend to send packets to the node. This requires coordination among the nodes.

Sparse Topology and Energy Management (STEM) [4] has been proposed to use a radio channel for wakeup signals and another data channel for data packet transmission where

the solution requires the nodes to equipment two different radio modules. Each node activates its wakeup radio for every  $T$  seconds and then goes to the sleep mode periodically. If a node intends to send a packet to a target node, it sends a stream of wakeup beacon in the wakeup channel to the target node. The drawback of this solution is, however, the introduction of another radio (wakeup radio) on the sensor nodes which consumes additional power. Moreover, the sending node needs to keep sending the wakeup beacon if the target node does not receive the beacon and that results in the sending node consuming more energy and increasing latency. Another solution is using a low power radio for the wakeup channel in order to save energy. However, the different transmission ranges between the two channels will cause a problem that a node is unable to wake up the target node even though these two nodes are within the data transmission range.

Sensor-MAC (S-MAC) [5] is another protocol designed for reducing the power consumption of sensor nodes. The transmission schedule of each node is divided into two parts: listening and sleeping. All nodes are synchronized and periodically wakeup and sleep. The Request-to-Send/Clear-to-Send (RTS/CTS) mechanism is used in S-MAC to avoid collision and hidden node problem. During the listening period, a sender node will send a RTS and wait for the CTS reply from the target node. If the sender receives the CTS reply, the sender starts to send data. Else, the sender node assumes that the RTS is not received by the target node and it goes to sleep until the next listening period. Any other nodes that overhear RTS or CTS will immediately go to sleep and wake up for a short period of time after the end of the transmission time. If an overhearing node is the next hop of the data packet, its neighbor can transmit the data to it without waiting for the next listening period. In S-MAC, however, nodes will suffer for the sleep latency due to the packet can only be forwarded during the nodes are active, and such latency may accumulate from one hop to another. Moreover, all nodes wake up at the same time and start to contend with each other by sending RTS, this may lead to the collision of RTS signals, thus resulting in low throughput.

DMAC [6] is designed to improve S-MAC by reducing the latency. DMAC uses the staggered wakeup pattern to create a pipeline for forwarding data in order to reduce the latency during data collection. That is, each node will have its own wakeup schedule and the part of its active period will overlap with other nodes. Therefore, the nodes are able to communicate during the overlapped wakeup time. The first advantage of this scheme is that all the nodes in the network may not need to be active at the same time, thus, the number of collision can be reduced. Secondly, each of node is able to save energy due to the node is only active for a short period. Thirdly, because of the staggered wakeup pattern, or the pipeline model, it reduces the packet forward latency. However, the pipeline model is effective only when each node has a single parent, In this case if a parent node fails, the rest of the subsequent nodes will be disconnected from the sink. On the other hand, it remains unclear how the

staggered wakeup time can be scheduled if a node is allowed to have multiple parents and multiple children.

In paper [7], the authors use a multi-parent approach to create multiple pipelines for forwarding packets. When parent nodes have simultaneous wakeup times, then one of the parent nodes will re-schedule its wakeup time. For example, in Figure 3, the parent nodes (nodes B and C) of node A have the same wakeup time, then node B will re-schedule its wakeup time. Thus, node A will have two time slots to send packets. If the node fails to send a packet during the current time slot, it can try to resend the packet for the next time slot without waiting to the next cycle. However, this protocol uses a fixed schedule and it is unable to adapt to the change of the traffic load. If the traffic load is high, nodes may not be able to transmit all the packets within the current active period and eventually it leads to increase the latency.

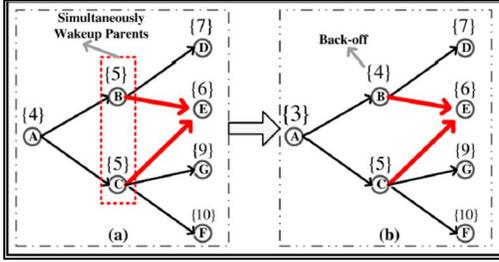


Fig. 3. Simultaneous Wakeup Parents [7]

### III. MAIN DESIGN

Each duty cycle of our protocol contains two phases, namely Slot Assignment and Data Transmission. Firstly, Slot Assignment is performed in order to schedule the activity of each node in each time slot in the Data Transmission phase. Note that there are three possible activities to be scheduled in each slot for a node, which are download, upload, and sleep. Coordination between nodes at two neighboring levels is required as when a child node is uploading, the corresponding parent node has to be in the download mode. In the Data Transmission phase, on the other hand, time is divided into a fixed number of, say  $m$  frames, and each frame contains a fixed number of, say  $n$ , slots. The duration of a slot is just enough for the transmission of a packet from one node to the next. In order to prevent interference, a multi-channel approach is used in which each node has a receiving frequency channel that is different from all neighboring nodes. All nodes will then follow the schedule determined in the Slot Assignment phase to perform activities in each slot in a frame, and the frame pattern will repeat  $m$  times until this duty cycle ends and the next begins. The details of the Slot Assignment and Data Transmission phases are discussed in the following subsections.

#### A. Distributed Slot Assignment Algorithm

A node may have multiple child nodes. If two or more child nodes are trying to send a packet to the same parent node at the same time, then a collision will happen. Therefore, a schedule for each child node to send packets

to its parent node is needed. However, it is not encouraged for a parent node to give a fixed schedule to each of its child nodes because not all child nodes have the same amount of packets to send in most of the time. Thus, an algorithm which allows each node to have adaptive uploading slots based on the traffic load is required.

Figure 4 shows a duty cycle in the designed protocol on each node for two neighboring levels, say levels  $L$  and  $L+1$ . Each node has its own duty cycle and it is divided into two phases. In the Slot Assignment phase, there are two sub-phases, namely Iterative Matching - Parent (IM-P) and Iterative Matching - Child (IM-C), in which the iterative matching proposed in [8] is modified to perform slot assignment. The main purpose of iterative matching is to match pairs of sender and receiver among the nodes in a particular slot. Since a node can play a role as a parent in a slot, but as a child in another slot, it is therefore necessary to have two slot assignment subphases for each node. Figure 5 shows the details of an iterative matching subphase, which consists of three steps: R - Request, G - Grant, and A - Accept. The function of each step is described as follows:

**Request:** A child node will broadcast a request message to all of its parent nodes. This request message consists of the number of slots needed. If the child node has more packets to send, then it can request for more slots.

**Grant:** A parent node may receive request messages from different child nodes. In this step, the parent node will choose its available slots and reply grant messages to the each of its child nodes. The grant message includes the information of the sender ID, the receiver ID and the slots for the child nodes to send their packets.

**Accept:** Child nodes may have received Grants from more than one parent node. Thus, the child node is required to select a parent in a slot and send an accept message to the parent. After the parent received the accept message, then the sender and the receiver are paired in that particular slot.

In an iteration of RGA steps, it is possible that a node is unable to schedule a slot for uploading/downloading a packet even if its parent/child has a common inactive slot, owing to packet loss and/or selection mismatch. Therefore, each node will repeat three times of RGA steps within one iterative matching subphase. Besides, each node has two iterative matching subphases which is the iterative matching as parent (IM-P) and the iterative matching as child (IM-C). All the same level nodes will be in the same phase. If level  $L$  nodes are in IM-P phase, then all of the nodes in level  $L+1$  will be in IM-C phase. All the nodes which are in the IM-C phase will start to broadcast a request message and then follow the iterative matching steps which is to wait for a grant message and reply with an accept message to targeted parent node. After the current iterative matching phase is completed, then the nodes will switch the mode of iterative matching. For instance, if the current iterative matching phase is IM-P then next phase will be switch to IM-C.

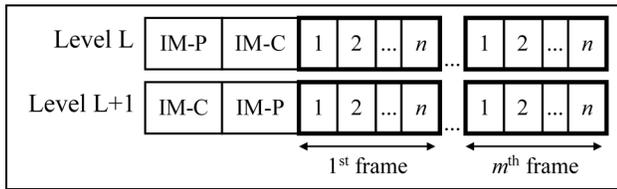


Fig. 4. A duty cycle in the protocol.

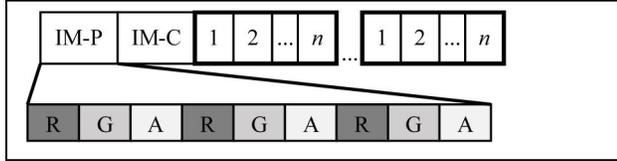


Fig. 5. Details of Iterative Matching Phase.

Figure 6 shows the duty cycle of each node. Let say node A is at level L, node B is the child of node A and node C is the child of node B. When the cycle starts, each node is required to perform the iterative matching. Node A and C are in the IM-P phase and node B is in the IM-C phase. Node B will start to broadcast a request message to its parent node. Assume that node B requested for two slots, then node A will select two available slots (for example, slot 1 and slot 2) and inform node B by sending a grant message. Node B will reply an accept message to node A if node B accepts the slots that provided by node A. After node A received the accept message, then node A and B are paired at slot 1 and slot 2 as shown in Figure 7. Node A will perform download and node B will perform upload at slot 1 and slot 2. Nodes can turn off its' radio for the rest of idle slots in order to save the energy. After the first iterative matching phase is completed, then each level of nodes will need to switch the mode of iterative matching to do the second iterative matching. For example, node A will switch to IM-C phase to do the iterative matching with its parent nodes (Level L-1). Node B will be in IM-P phase which will do the pairing with its child node (e.g. Node C).

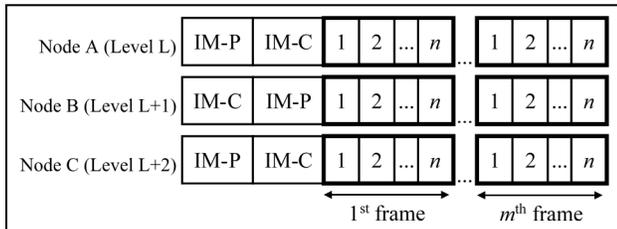


Fig. 6. Duty Cycle of each node

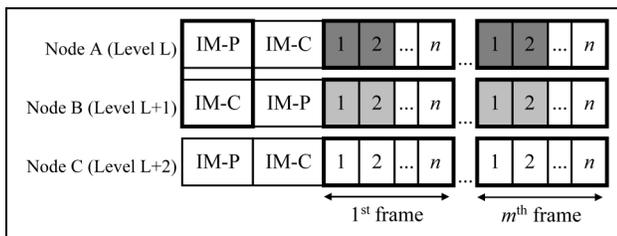


Fig. 7. Pairing between Node A and B

### B. Multi-channel Data Transmission

Slot assignment avoids collision caused by child nodes send packet to the same parent simultaneously. However, collision will still happen when a node sends a packet to the receiver, all of its neighbor nodes can actually receive it, so called overhearing. In additional, the hidden node problem will cause collision at the receiver. For example, in the Figure 8, when node B is sending a packet to node A, node C will overhear the packet. If node D tries to send a packet at this time, then collision will be happened at node C resulted that the packet did not receive by node C correctly. Thus, node D is required to retransmit the packet again.

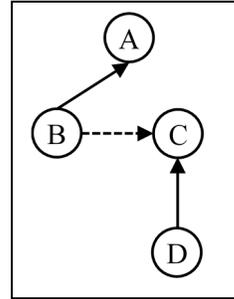


Fig. 8. Overhearing and collision at node C

Therefore, multi-channel approach is performed to distribute other neighbor nodes in order to avoid from collision and overhearing. First of all, all the nodes will have its own selected channel. This channel cannot be same as the neighbor nodes' channel and also the channel of the parent of neighbor nodes. All the child nodes are able to know the channel of its parent node. When the child node wants to send a packet to its parent node, then it will switch its channel to the parent node's channel. After that, child node can start to send a packet to the parent node and the parent node will able to receive it. Figure 9 illustrates a scenario of multi-channel usage during data transmission. Each node will switch channel to its parent's channel during data transmission and avoid interference from other nodes. After the data are sent by the sender, it will switch back to its own channel.

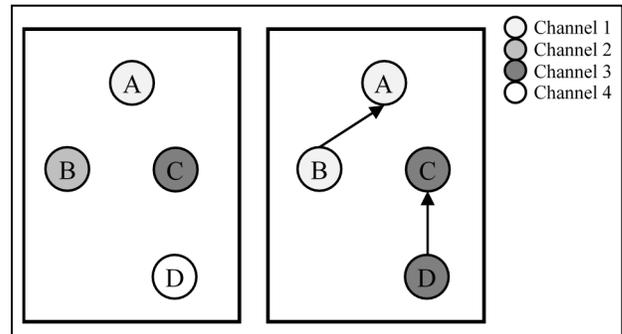


Fig. 9. Multi-channel during data transmission

The combination of multi-channel division and slot allocation is able to provide a collision free data transmission. Multi-channel division avoids interference from each other node and slot allocation prevents multiple

child transmit data to the same parent node simultaneously. Since the channel is free for the data transmission, a backoff before sending data is not required to perform. Therefore, the slot duration can be trimmed to the duration of sending one data plus the time of waiting for ACK packet.

#### IV. SIMULATION RESULT

In this section, the simulation is carried out by using MiXiM [9] which is one of the OMNeT++ modelling frameworks. The simulation setting is shown as in Table 1. The parameters of current drawn in transmitting, receiving, idle and sleep modes are referred to CC2420 datasheet [10]. The metrics of energy consumption, packet loss, latency, and the number of transmission are selected for the comparison of the proposed protocol against multi-pipeline scheduling [8] and the proposed multi-path with single channel protocol (MS) which is using the standard of IEEE 802.15.4 CSMA/CA protocol without any sleep/wake protocol. In additional, we focus on level-1 nodes because level-1 nodes have higher impact to the rest level of nodes due to if all of the level-1 nodes are ran out of their energy, the rest will be disconnected directly from the base station. In the simulation, 100 nodes are generated and randomly distributed. The network topology structure is shown as in Figure 10. All the nodes will periodically generate packets and send to the base station through multi-hop and multipath. We assume all nodes transmit the same type of data packets. The duration of one cycle is configured to 4.572 seconds for our protocol with multi-pipeline scheduling. In our protocol, the duration of 10% in the cycle is used to perform the iterative matching process, the rest of time is used for data transmission which consist of 86 frames. Each frame contains 16 slots with 0.003 seconds. In the meantime, the cycle of multi-pipeline scheduling is divided into four frames. Each frame consists of 16 slots with 0.072 seconds.

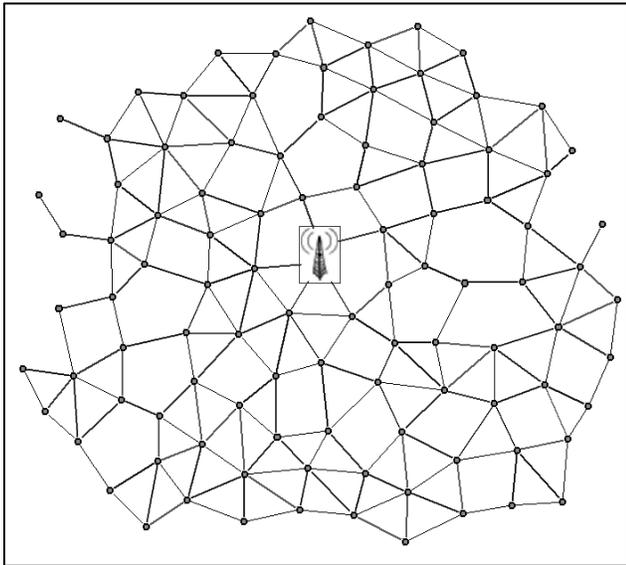


Fig. 10. Network topology structure

TABLE I. SIMULATION SETTING

Parameter	Value	Description
Network Size	100	Number of nodes in network
Packet Size	64 bytes	The size of the data packet
BER	0.01%	Bit error rate
Radio Range	50m	Radio transmission range
Bit Rate	250kbps	Transmit bit rate
TX current	17.4 mA	Current drawn during transmitting packet
RX current	18.8 mA	Current drawn during receiving packet
Idle current	0.426 mA	Current drawn when radio is idle
Sleep current	0.021uA	Current drawn when radio is switched off

The result in Figure 11 shows the average energy consumption in level-1 nodes. From the figure, the nodes which do not implement sleep protocol consumed the most energy. Multi-pipeline scheduling has the lowest energy consumption even when the traffic load is increased. This is because multi-pipeline uses a fixed scheduling and the sleeping time is constant. However, our protocol will have higher energy consumption than multi-pipeline scheduling. The reason is that during the high traffic load, our protocol is able to convert some of the sleeping time into active time to transmit or receive packets.

Figure 12 shows the end-to-end packet loss under different traffic load. Multi-pipeline scheduling has highest packet loss rate where our protocol has lowest packet loss rate. Due to the multi-pipeline scheduling uses fixed schedule, the outgoing traffic load of each node is limited. Therefore, when the traffic load is getting higher, the nodes will start to discard packets and lead to increase the packet loss rate. For the normal scheme (without sleep protocol), when the traffic load more than 1.5, packet loss rate is started to increase because each node is reached the maximum throughput. Thus, further increasing the traffic load will lead to higher packet loss rate. Our protocol is able to maintain the packet loss rate near to 0 even when the traffic load is more than 1.5.

The results of end-to-end latency are shown in Figure 13, Figure 14 and Figure 15 with level-1, level-3 and level-6 nodes respectively. Level-6 nodes are the most far away from the sink in this network topology structure. Therefore, the latency in level-6 nodes will be higher than other levels. During the low traffic load, MS protocol has lowest latency compares to others. It is because each node can immediately transmit packet to its parent when it received or generated a packet. In our protocol and multi-pipeline scheduling, the latency is higher than MS protocol due to the slot allocation. Each node has to wait until the active time then can only transmit packet to parent node. From the figure, the latency of multi-pipeline scheduling increased dramatically from 0.4 to 0.5. Besides that, in the MS protocol, the latency starts to increase when the traffic load more than 1.5. The reason is because when the incoming traffic is higher than outgoing traffic, the packet will be queue in the buffer. The queuing delay resulted high latency.

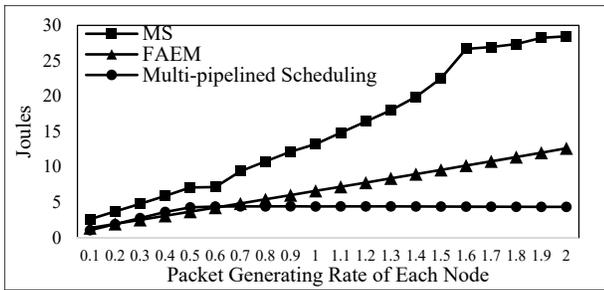


Fig. 11 Average Energy Consumption in Level-1 Nodes

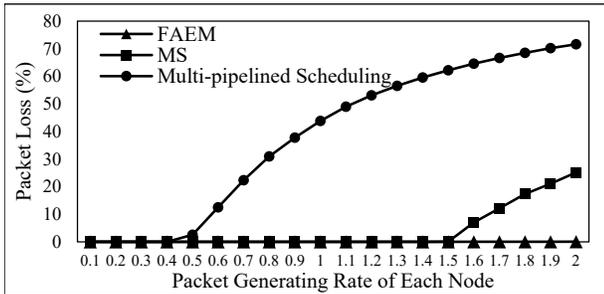


Fig. 12. Packet loss rate vs Traffic load

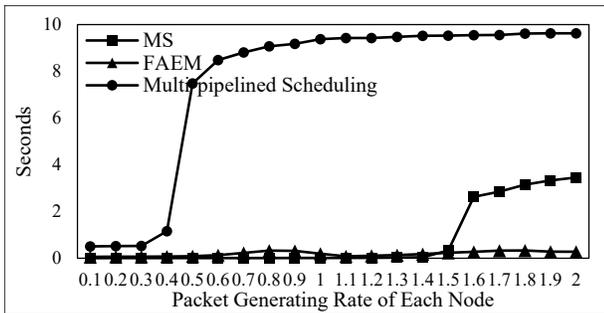


Fig. 13. Average Latency in Level-1 Nodes

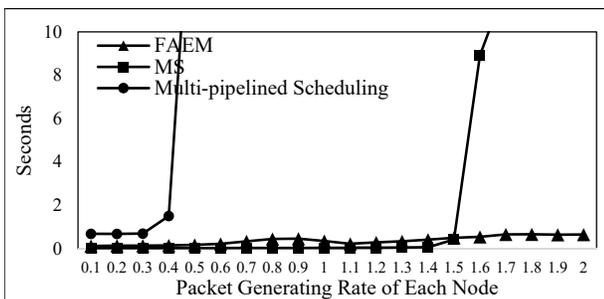


Fig. 14. Average Latency in Level-3 Nodes

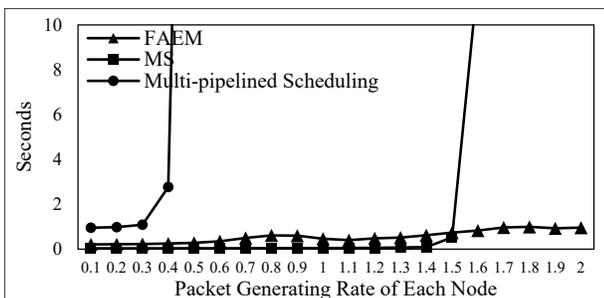


Fig. 15. Average Latency in Level-6 Nodes

## V. CONCLUSION

Our protocol is able to achieve the lower energy consumption. Thus, it can prolong the network lifetime. Besides that, our protocol does not use a fixed scheduling. An adaptive scheduling based on the traffic load allows our protocol to achieve the low latency during the high traffic load. In additional, the combination of slot allocation and multi-channel division provides a collision free data transmission result that each of the node can directly send out packet without waiting a backoff period and sensing channel. Thus, the outgoing traffic load of each node is increased result that our protocol is able to achieve data reliability in high traffic load. Other performances such as memory size and throughput will be study in feature work.

## ACKNOWLEDGMENTS

This research work is supported by Newton-Ungku Omar Fund, British Council under Project ID: 172704569.

## REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", in Communications magazine IEEE, vol. 40, pp. 102–114, 2002.
- [2] D. Goyal, M. -R. Tripathy, "Routing Protocols in Wireless Sensor Networks: A Survey", in 2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT), Rohtak, Haryana, 7-8 Jan 2012, pp. 474 – 480.
- [3] S. -Y. Liew, M. -L. Gan, C. -S. Lim and H. -G. Goh, "Basketball Net – A Flexible and Resilient Topology for Wireless Sensor Networks", in Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8-11 July 2014, pp. 87-92.
- [4] C. Schurgers, V. Tsiatsis and M.B. Srivastava, "STEM: Topology management for energy efficient sensor networks", in Aerospace Conference Proceedings, Big Sky, MT, 10-15 March 2002, vol. 3, pp. 1099-1108.
- [5] W. Ye, J. Heidemann, D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", in IEEE/ACM Transactions on Networking (TON), vol. 12, pp. 493–506, June 2004.
- [6] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An adaptive energyefficient and low-latency Mac for data gathering in wireless sensor networks", in: Proc. 18th International Parallel and Distributed Processing Symposium, April 2004, pp. 224-231.
- [7] Y. Cao, S. Guo and T. He, "Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks", in: INFOCOM, 2012 Proceedings IEEE, Orlando Florida, 25-30 March 2012, pp. 361-369.
- [8] D. Stiliadis, A. Varma, "Providing bandwidth guarantees in an input-buffered crossbar switch", in: INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE, Boston, MA, 2-6 April 1995, vol. 3, pp. 360-368.
- [9] Mixim Project [online]. Available at: <http://mixim.sourceforge.net>
- [10] 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver [online]. Available at: <http://www.chipfind.net>