2011 IFIP Ninth International Conference on Embedded and Ubiquitous Computing

IMAC: An Interference-aware Duty-cycle MAC Protocol for Wireless Sensor Networks Employing Multipath Routing

Leila Eskandari*, Hamed Yousefi†, Ali Movaghar†, and Mohammad Khansari‡
*Department of Information Technology, Sharif University of Technology, International Campus, Kish, Iran
leila_eskandari@kish.sharif.edu
†Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
hyousefi@ce.sharif.edu, movaghar@sharif.edu
‡Iran Telecommunications Research Center, Tehran, Iran
khansari@itrc.ac.ir

***Abstract*—The main source of energy consumption in the current MAC protocols for wireless sensor networks is idle listening. To mitigate this problem, duty cycling is used. However, it increases data delivery latency. In this paper, we propose an Interference-aware duty-cycle MAC (IMAC) protocol for wireless sensor networks that uses cross-layer information to reserve multiple paths for each source and send data packets along them efficiently. IMAC also handles the existing interference between these paths such that data packets can be delivered in the minimum required number of cycles. Simulation results in ns-2 show that the proposed algorithm has an average reduction of 49% in data delivery latency compared to a current solution called RMAC.**

***Keywords*-Wireless Sensor Network; Cross-layer Design; Duty-cycle MAC; Multipath Routing.**

I. Introduction

The development of a small single node with the capability of sensing phenomena from the surrounding environment, data processing, and communicating is the result of recent advances in Micro-Electro-Mechanical Systems (MEMS) technology. A sensor network is composed of a large number of sensor nodes which are densely deployed either inside the phenomenon or very close to it and cooperate to create a sensing application. Sensor nodes are energy constrained due to limited and irreplaceable batteries [1]. Therefore, all designed layers in the architecture of Wireless Sensor Networks (WSNs) must be energy efficient. The main source of energy consumption in the MAC layer is idle listening [2]. In many traditional MAC protocols for wireless networks, nodes remain awake to send and receive data. However, this wastes significant energy in WSNs where nodes are energy constrained and the traffic load is usually light. To mitigate this, duty cycling is a very important method used to switch nodes between being awake and sleeping. Thus, in an active mode, a node sends a control packet if it wants to send data or listens to the media if there is a data packet destined to it. In a sleep mode, after sending or receiving data, nodes go to sleep to save energy. However, the use of duty cycling increases the data delivery latency, especially in multi-hop WSNs. Each node sends a data packet one hop further and the next hop should wait for the next operational cycle to forward it. Some protocols can forward the data packets multi hops away to overcome this latency. For example in S-MAC with adaptive listening [3], a packet is forwarded two hops away in a cycle. In RMAC [4], a node even forwards the data packets more than two hops. By the use of a special control packet, a data path is reserved in each cycle. If a node along the path receives a control packet destined to it, it sets the correct wakeup time to receive and forward data and sleeps during the other times.

In RMAC each source only sends one packet per cycle. However, it suffers from the data delivery latency when the packet generation interval is less than the duration of a cycle. In this paper, we propose an Interference-aware duty-cycle MAC (IMAC) protocol. Here, by employing multipath routing in RMAC and sending out multiple packets over multiple paths, we improve RMAC significantly. IMAC can schedule data flow along each path without interfering with the other paths. In other words, the interference-aware feature of the new protocol ensures that the interference between these paths is removed as much as possible. It can be done by changing wakeup time of the node that overhears a transmission and is not allowed to send its data in that period of time. IMAC lets this node try sending its packet when the interference is removed from the contention area.

The reminder of this paper is organized as follows. In Section II some related works for reducing data latency in the duty cycle based MAC protocols are discussed. Section III describes the proposed algorithm in details. Section IV presents the simulation results, including a comparison of IMAC with RMAC. Finally, Section V concludes the paper.

II. Related Work

A number of duty-cycle based MAC protocols have been proposed in the literature. S-MAC [5] is a well-known MAC protocol for WSNs which uses duty cycling. It saves energy by using periodic active/sleep. Each cycle has three periods: SYNC, DATA, and SLEEP. Nodes wake up at the beginning of the SYNC period and synchronize their clocks with each

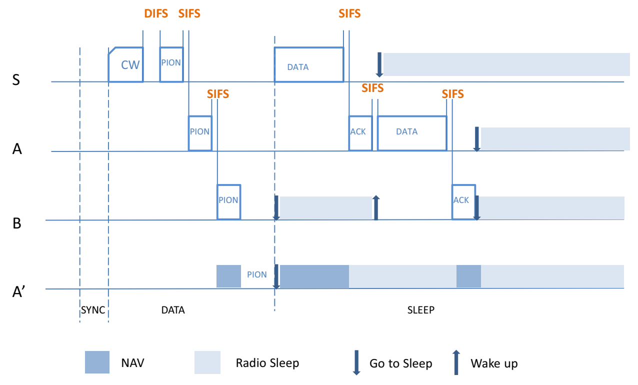

Figure 1. Overview of RMAC

other. During the DATA period, all nodes should stay awake to send and receive RTS and CTS control packets. Nodes that are involved in a transmission, send and receive data packets during the SLEEP period and then sleep, others go to sleep immediately at the beginning of the SLEEP period to save energy. In S-MAC with adaptive listening [3], a data packet is delivered two hops away in a single operational cycle.

RMAC [4] tries to improve S-MAC, send a data packet multi hops away, and reduce data latency. Like S-MAC, RMAC cycle has three periods. It uses some cross-layer information including final destination and the number of hops that a PION has traveled by using a counter and sends a setup control frame called PION to schedule the upcoming data packet delivery along the path during the DATA period. A PION is used to request communication like an RTS frame and confirm a request like a CTS frame. During the SLEEP period, a node goes to sleep and wakes up when the upstream node transmits a data packet destined to it and then forwards the data packet to the next hop. Thus, the wakeup time of ith node along the reserved path is calculated as follows:

$$T_{wakeup}(i) = (i-1).(durDATA+SIFS+durACK+SIFS) \tag{1}$$

Where *dur*DATA and *dur*ACK are the required time to send a data packet and an ACK, respectively. Figure 1 shows an overview of RMAC protocol. Node A′ is a neighbor of node A and should not transmit when node A is receiving anything to avoid collision. Therefore, node A′ sets its NAV in three segments of time when it overhears a PION transmission from A to B. Node A′ should not send PION when node A is receiving PION confirmation from B. Moreover, it should not transmit data when node A is receiving data from S or ACK from B [4].

RMAC delivers one data packet per cycle, so data delivery latency is increased in the high traffic load. There are some protocols based on RMAC, like PRMAC [6], D-RMAC [7], BulkMAC [8] that try to improve RMAC and adapt it to the burst traffic.

PRMAC [6] enables multiple packet transmission in a flow. Every node informs its next hop the number of packets it is going to send including the number of packets to be received from the previous hop for this flow. Each node waits for the minimum time duration that a node should postpone sending new packet so that the previous packet gets far enough and does not collide with the new one. Comparing to RMAC, PRMAC's PION has some more cross-layer information including the number of data packets that a node wants to send to the downstream node.

D-RMAC [7] improves latency in RMAC by dynamically adjusting the duty cycle according to the current traffic condition. In RMAC, the duty cycle is fixed and when it is smaller than the packet generation interval, transmission delay severely increases. To solve this problem, when the traffic load increases, a source node doubles its duty cycle, includes it in the PION frame, and sends it to the next hop. When the next hop receives PION, it updates its duty cycle. As the traffic load decreases, the source node changes the duty cycle by half and informs the next hop via PION frame, so the duty cycle is adapted based on the traffic load of the network.

BulkMAC [8] works similar to RMAC in the low traffic load, but as the traffic load increases, it delivers multiple data packets in a cycle. It has two operation modes. In the single hop multiple receiver mode, transmission of multiple packets towards different destinations is scheduled while in the multi hop flow mode, a packet is transmitted over multiple hops. In the former mode, the sender transmits a Single Hop Multiple Receiver Frame (SHMRF) to the destinations during the DATA period. Comparing to the RMAC PION, this frame has multiple different destinations and the number of packets to each of them. In the latter mode, a Multi-Hop Frame (MHF) is used which includes send start and end start indices in addition to RMAC PION frame. These indices tell each intermediate node when it is going to receive data packets from the previous node.

III. Protocol Overview And Properties

As mentioned in the previous section, RMAC cannot send multiple data packets from one source in one cycle. Discussed protocols PRMAC, D-RMAC, and BulkMAC improve data delivery latency of RMAC. PRMAC and Bulk-MAC send multiple packets in a cycle, but they send them along the same path and it causes that the remaining energy of the nodes in the path decrease significantly compared to the other nodes. In other words, energy consumption of the nodes is not uniformly distributed. Some nodes may die sooner than the others. In addition, they calculate the correct wakeup time of a node based on the hop count from source to that node. However, when this hop count for two nodes from two different paths is the same, only one of them can send data and the other node should wait for the next

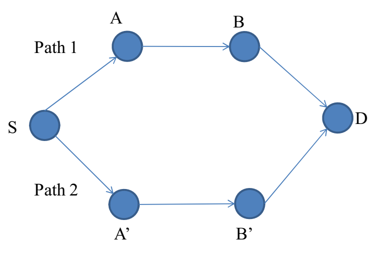

Figure 2. Two packets are sent out along two paths from S to D

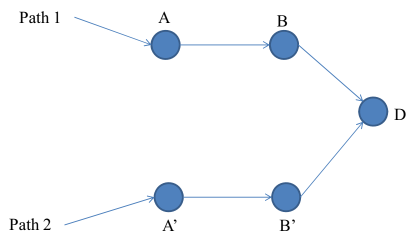

Figure 3. Two paths interfere with each other around the sink

cycle. This increases the data delivery latency. IMAC tries to solve these problems by sending the data packets along multiple paths. It also can handle the interference between these paths by using the information that a node obtains from overhearing during the DATA period. In IMAC, if a source node has multiple packets to send to the destination, it initiates communication request along multiple paths which are provided by multipath routing and sends each data packet from one path.

Each cycle in IMAC is divided into three periods: SYNC, DATA, and SLEEP. During the SYNC period, similar to RMAC, nodes synchronize their clock with each other. However, the DATA and SLEEP periods differ from RMAC and are described in details in the following subsections.

A. Data Period

The Data period in IMAC is different from RMAC becausea source node should send out more than one PION. It is also different when two paths interfere with each other. In addition to cross-layer information of RMAC, in IMAC, the multipath routing protocol provides the number of paths that a source can send its data packets along for the MAC layer. This number may not be equal to the number of packets to be sent. When it is more, the number of PIONs sent out from source is equal to the number of packets and when it is less, the number of PIONs is equal to the number of paths. We consider two examples to see IMAC operation. At the first one, as shown in Figure 2, node S has two data packets and the number of paths that multipath routing protocol has provided is two. Hence, it sends out two PIONs along two available paths. Since each PION is overheard by its sender when it gets relayed, source S cannot send two PIONs simultaneously. It should wait until the first PION gets far enough and does not interfere with the second one. This time can be calculated by the following equation:

$$T_d = \alpha.\lceil \varphi / r \rceil.(SIFS + durPION) \tag{2}$$

Where α counts the number of times that a node tries to send data but overhears a transmission, φ is the interference range, r is the transmission range, and *dur*PION is the time required to transmit PION. After this time, the source can send its second PION. In this case, suppose that source S sends its first PIONs along path 1 to node A. Then node A sends PION to node B, this transmission is overheard by S as acknowledgement of receiving PION by node A. After T_d node S can send its second PION to node A$'$ along path 2.

The second example of managing interference is shown in Figure 3 where two different paths from two different sources interfere with each other around the sink. In RMAC, when two nodes are in the same transmission range, only one of them can send its data and the other one should set its NAV and wait for the next operational cycle to send its data. However, IMAC can handle this interference scheme and both sources can relay their data in one cycle.

Nodes A and A$'$ are at the same transmission range, so they cannot send their PIONs to B and B$'$ simultaneously. Only one of them, for example A, wins the contention and can send PION to its next node, B. Node A$'$ overhears transmission of PION from A to B and finds out that it cannot transmit its PION at that time and should set its NAV, but what about the time after that when node A is quiet? The time duration that node A$'$ has to wait to relay its PION is also calculated by (2). Thus, node A$'$ sends its PION as soon as the contention is removed from the contention area.

B. SLEEP Period

In the Sleep period each node wakes up at the correct time and receives upcoming data and then sends it to the downstream node. As mentioned before, IMAC takes into account the number of times that a node cannot transmit its data and hop count to calculate the correct wakeup time as follows:

$$T_S = (\alpha + \beta - 1).(durDATA + SIFS + durACK + SIFS) \tag{3}$$

Where β is the number of hop that a packet has traveled from source so far.

Consider the first example in Figure 2 when source S has two data packets to send to destination D. The DATA and SLEEP periods for path 1 and path 2 are shown in Figure 4 and Figure 5, respectively.

We also prepare the DATA and SLEEP periods of the second example. Node A wins the contention, relays its

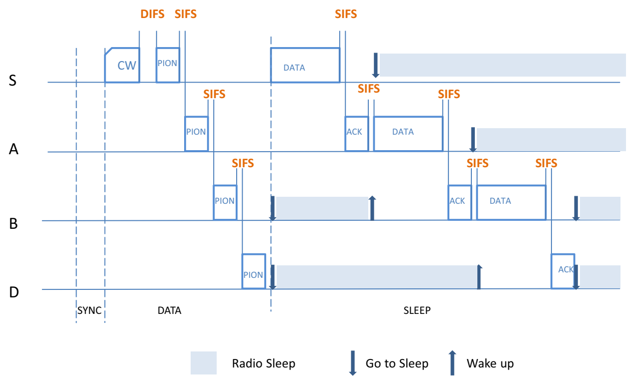

Figure 4. IMAC cycle for path 1 at the first example

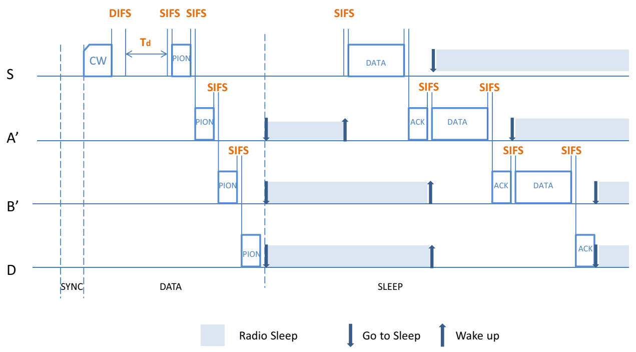

Figure 5. IMAC cycle for path 2 at the first example

PION and calculates its wakeup time based on (3). α is zero for node A since it has not overheard any other transmission. Node A′ overhears that node A wants to send data at the time that it was going to. Therefore, it adds one to α . Figure 6 and Figure 7 show the operation of the nodes along path 1 and path 2, respectively. The point S is the beginning of the contention area. Node A′ waits for T_d and tries again to send its PION to its next hop at the same cycle when first PION is far enough and there is no contention. The point W in Figure 7 is when node D wakes up to receive data from B.

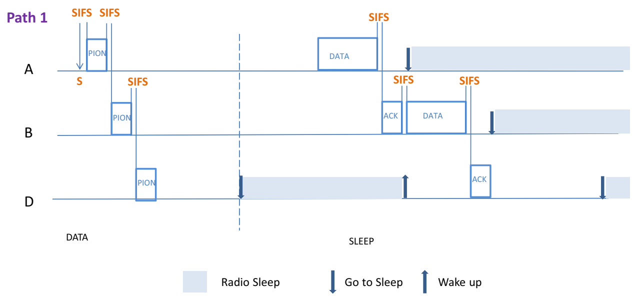

Figure 6. IMAC cycle for path 1 at the second example

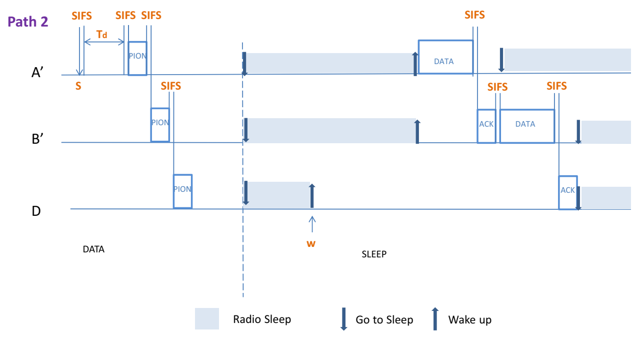

Figure 7. IMAC cycle for path 2 at the second example

IV. Performance Evaluation

We evaluate our proposed algorithm, IMAC, using version of 2.29 of ns-2 [9]. In simulation, we assume that the nodes have one schedule. Moreover, the routing information is provided for the MAC layer. Table I lists the parameters used in simulation which is the same as the parameters in RMAC [4].

We use three scenarios to compare our method with RMAC. First, we consider two chains which are neighbors and each of them has one source node as shown in Figure 8. All nodes in each chainare 200 m apart. One single CBR (Constant Bit Rate) flow sends packets from each source to the sink. Each chain includes at most 24 hops, including the sink node. This scenario is designed to compare the operation of the protocols in the existence of interference between two chains.

The cross scenario as shown in Figure 9 includes two chains that cross each other at the center. One CBR flows generates the traffic loads at each chain at the same time and at the same rate. Thus, their traffic contends with each other at the center. Each chain includes at most 24 hops as well. The goal in the cross scenario is to study that how IMAC can handle existing contention at the center of nodes compared to RMAC. Finally, in the realistic scenario, a WSN is composed of 100 randomly distributed sensor nodes in a 2000 m by 2000 m square area, and a node as the sink. The number of relayed PION in a cycle in all three scenarios is 8. In all these scenarios, the main performance metrics of interest are data delivery delay and energy consumption. Moreover, the duty cycle for all the scenarios is set to 5%.

A. Data Delivery Latency

To evaluate the end to end delay of IMAC protocol in two neighbor chains scenario, the path length is changed from one hop to 24 hops when each source has one data packet to send. Then, we measure the average packet delivery latency of two sources. In this scenario, we want to show that IMAC can handle interference between two sources. The results in Figure 10 show that there is a wide gap between RMAC and IMAC, except for a few points. It can be explained

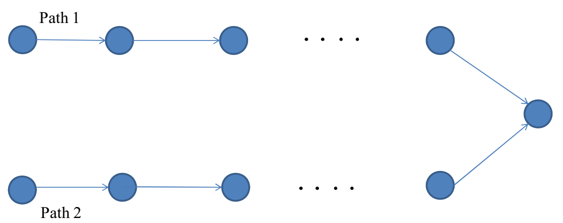

Figure 8. Two neighbor chains scenario

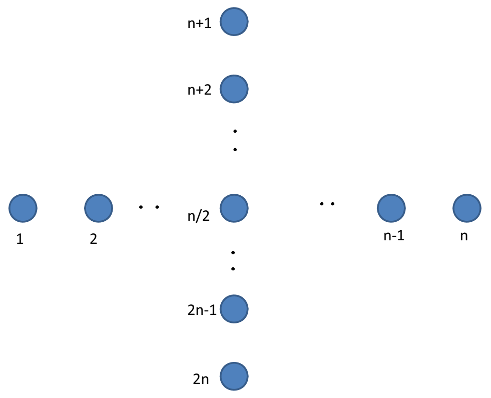

Figure 9. Cross scenario

as follows, because of the interference, the second source in RMAC has to wait for the next cycle to send its data. However, in IMAC, it can be sent at the same cycle when the first source has sent its data, so it saves one cycle. At some points there are a few jumps in IMAC and it works almost similar to RMAC. The reason is that second source sends some of its PIONs during the DATA period of the next cycle, so the number of cycles in RMAC and IMAC becomes the same. However, IMAC still outperforms RMAC because it has sent as many as possible PIONs in the previous cycle.

In the cross scenario, it is shown that IMAC can handle the existing contention at the center of the nodes and allow the second source to send its data as soon as the contention is removed from the contention area, not in the next cycle. As it can be seen in Figure 11, in the most points of the

Table I
SIMULATION PARAMETERS

Bandwidth	20 Kb/s
RX Power	0.5 W
TX Power	0.5 W
idle Power	0.45 W
Sleep Power	0.05 W
TX Range	250 m
CS range	550 m
CW	64 ms
DIFS	10 ms
SIFS	5 ms

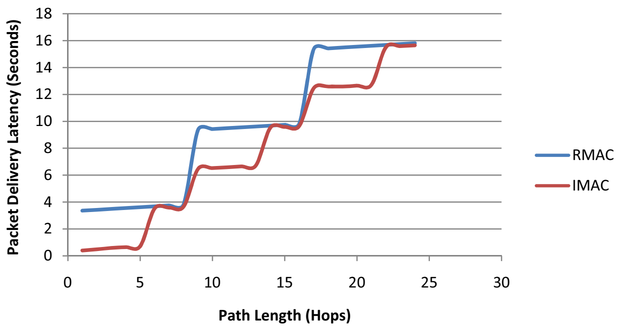

Figure 10. Delivery latency in two neighbor chains scenario

path length, IMAC performs better than RMAC. However, in a few points they work pretty much the same. The reason is that in those points the number of PIONs that can be sent in the DATA period has reached to its maximum value by the first source, so when the DATA period is finished, the contention is already removed from the contention area and both RMAC and IMAC perform the same after that. The key concept of IMAC is that it tries to send PION from the second source after removing contention at the same cycle, so it saves at least one cycle.

In the realistic scenario, we consider three sources at the farthest possible points from sink and each source has two packets to send. These sources are not at each other the same transmission range. Multipath routing provides two paths for each source. The paths from these sources to the sink interfere with each other. Path length is changed from 1 to 10 hops. In IMAC, each source sends its two data packets towards the sink via two different paths while in RMAC the data packets are sent via the same path. The packet delivery latency is calculated as the summation of delivery latency of all packets divided by the number of packets. The simulation results are provided in Figure 12. RMAC operates the same for hops 1 to 8, the reason can be explained as follows. A source cannot send two PIONs for its two data packets at the same cycle, and the number of PIONs that can get relayed in one cycle is 8. Moreover, these six paths interfere with each other around the sink. Thus, the number of cycles is the same for hops 1 to 8. There is a jump in path length 9, because sources have to send one PION in the next cycle. However, IMAC can send two PIONs from one source over two paths in one cycle. It can also handle the potential interference between these paths. Moreover, IMAC works very efficient around the sink. The simulation results as presented in Figure 12 show that IMAC has a significant improvement of 49% when source nodes are 10 hops away from sink in the realistic scenario.

B. Energy Consumption

To measure energy consumption in IMAC against RMAC, we consider the realistic scenario. Since in two neighbor

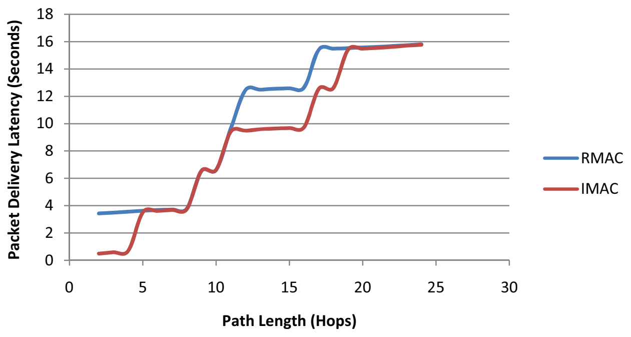

Figure 11. Delivery latency in the cross scenario

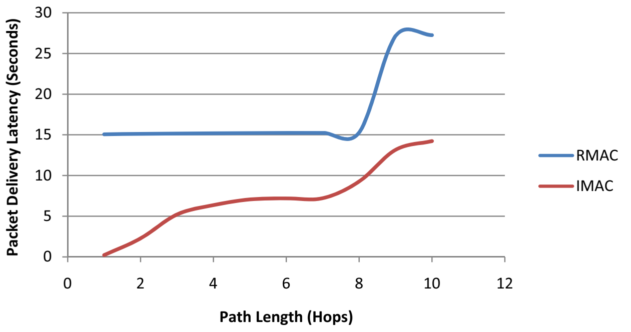

Figure 12. Delivery latency in the realistic scenario

chains and cross scenarios the number of saved cycles is maximum 1, there is not much difference between IMAC and RMAC energy consumption. The number of packets fed to network is changed from 0 to100. Each source sends two packets every 50 seconds. Average energy consumption is calculated by dividing total energy consumption of all sensors by the required time to send data packets. The results show that IMAC saves more energy than RMAC as can be seen in Figure 13. The reason can be explained as follows. Although the number of all the PION and DATA and ACK transmissions in both protocols is the same, IMAC reduces the number of cycles of the whole packet transmission.

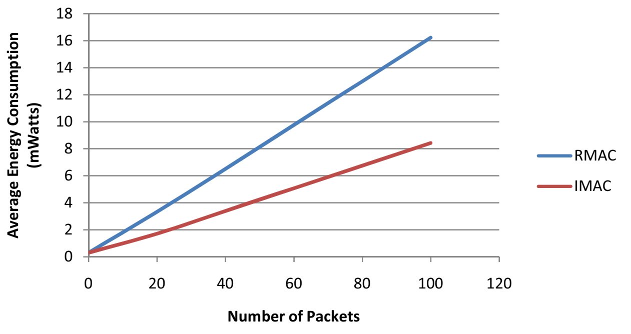

Figure 13. Energy consumption in the realistic scenario

V. Conclusion

In this paper, we proposed IMAC to improve the performance of wireless sensor networks. IMAC is inspired by RMAC which is a duty cycle based MAC protocol. Unlike RMAC, in IMAC each source can send more than one data packet per cycle by sending multiple PIONs along the multiple paths and handling the existing interference of these paths. To evaluate the performance of IMAC, we used ns-2 and compared data delivery latency and energy consumption of IMAC with RMAC. The simulation results show that IMAC outperforms RMAC and has a significant improvement. Numerical analysis of the average packet latency of IMAC in a probabilistic manner can be considered as a future work.

VI. Acknowledgement

The authors would like to thank Dr. Shu Du who provided them the implementation of RMAC generously.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] M. L. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," in *Proceedings of the 23th IEEE International Conference on Computer and Communications (INFOCOM'04)*, 2004, pp. 1740–1750.

[3] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[4] S. Du, A. K. Saha, and D. B. Johnson, "RMAC: A routing-enhanced duty-cycle mac protocol for wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer and Communications (INFOCOM'07)*, 2007, pp. 1478–1486.

[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 21th IEEE International Conference on Computer and Communications (INFOCOM'02)*, 2002, pp. 1567–1576.

[6] T. Canli and A. Khokhar, "PRMAC: Pipelined routing enhanced mac protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC'09)*, 2009, pp. 1–5.

[7] Y. I. Kim, B. Choi, C. S. Kim, and M. Y. Chung, "A synchronized mac protocol considering packet generation intervals in wireless sensor networks," in *IEEE Region 10 Conference (TENCON'09)*, 2009, pp. 1–5.

[8] T. Canli, M. Hefeida, and A. Khokhar, "BulkMAC: A cross-layer based mac protocol for wireless sensor networks," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC'10)*, 2010, pp. 442–446.

[9] "The network simulator, http://www.isi.edu/nsnam/ns."