Cross-layer Opportunistic Forwarding to Reduce Patterned Synchronization Effect in Highly Resource Constrained WSNs

Asanka Sayakkara* and Dongkyun Kim†
*asanka@monet.knu.ac.kr, †dongkyun@knu.ac.kr
School of Computer Science & Engineering,
Kyungpook National University, Korea

***Abstract*—In duty-cycled wireless sensor networks, senders suffer from long delays while waiting in awaken state for receivers to wakeup for packet transmissions. It results in unnecessary energy usages and lower network lifetime. Opportunistic packet forwarding schemes have been proposed to reduce such delays where the selection of next hop forwarder is performed opportunistically without waiting for a particular neighbor to wakeup. Even under the opportunistic nature of such schemes, the duty-cycle patterns of neighboring nodes can require a sender to be synchronized with a particular receiver so that it is always selected as the next hop forwarder. This effect, which is called as *patterned synchronization effect* in this paper, leads to unnecessary drain of energy in packet forwarder nodes reducing their lifetime. We therefore propose a new opportunistic forwarding scheme which is able to identify the occurrence of *patterned synchronization effect* and resolves it by dynamically adjusting duty-cycle ratio of nodes in a self-organized manner. Using extensive simulations, we show that our scheme significantly improves network lifetime and power utilization of the network while maintaining an acceptable packet transmission delay at the senders.**

I. Introduction

Wireless sensor networks (WSNs) are composed of battery-powered nodes. Hence, much research work has paid attention to saving energy in sensor nodes. In particular, sensors waste their energy significantly even during idle listening. In addition, the amount of energy consumed for a packet reception is nearly equal to that of energy consumed in idle listening for a similar time duration [9], [1]. Recently, *radio duty-cycling* mechanisms have been proposed to reduce energy consumption by requiring sensor nodes to periodically turn on and off their radio transceivers from the MAC layer. Such MAC schemes can be categorized as *synchronous* and *asynchronous*. In the former, all the nodes in the network have synchronized duty cycles, while each node can have its own duty-cycle schedule in the latter. Synchronous MAC schemes have some drawback in that the synchronization of the whole network itself consumes extra energy despite no data transmissions in the network. Therefore, asynchronous MAC schemes are considered more suitable for WSNs.

One-hop transmissions at the MAC layer can be performed after the potential next-hop neighbor towards the sink is decided from a routing protocol at the network layer. However, due to the nature of asynchronous duty-cycle MAC schemes, a sender always has to wait and consume its energy until the selected next-hop receiver wakes up from the duty-cycle. To minimize the energy waste in waiting, cross-layer opportunistic routing schemes have been proposed to provide multiple options for the next hop neighbor decision. In such schemes, the first awaken neighbor from a pool of potential candidate nodes (rather than one designated neighbor) is selected as the receiver [6]. Any neighbor which is having a route towards the destination can be a potential candidate for the next hop. Most practical WSN applications consist of a routing tree where the sink is positioned at the root [4]. All the data packets generated by sensor nodes in the network are supposed to be routed towards the root of the routing tree, where all the nodes in the network have at least a single route towards the sink. In a dense network where each node has many neighbors, cross-layer opportunistic routing schemes are applicable to reduce such a waiting delay. Based on the information of MAC layer regarding the moments where neighbors wake up, routing layer can make immediate decisions for the suitable next hop within a minimum delay.

However, we found out a unique problem that most cross-layer opportunistic forwarding schemes overlooked. When data is generated at a constant rate and nodes have a constant duty-cycle ratio, the sender will select the same receiver all the time as its next-hop. In other words, due to the duty-cycle patterns in neighbors, whenever a node is ready to send a packet, a particular neighbor may always wake up from the duty-cycle. This causes such a receiver to participate in packet forwarding too frequently than other suitable next-hop nodes. We refer to this phenomena as *patterned synchronization effect* in the rest of the paper. When this effect occurs, the traffic load from a sender towards the sink may always tend to traverse through the same route under the opportunistic scheme. It results in an uneven traffic distribution over the network causing the packet forwarders in such routes to run out of their battery power too early. Since achieving as longer network life as possible is a critical requirement in WSN, *patterned synchronization effect* can be considered as a serious challenge. Existing cross-layer opportunistic schemes in the literature for duty-cycled WSNs which are designed to reduce the waiting delay at senders are unable to detect the occurrence of this effect. Therefore they can suffer from sudden node failures by running out of battery shortening their network lifespan.

In this paper, we therefore propose a novel cross-layer opportunistic forwarding scheme which minimizes the waiting delays at senders, thus leading to a longer network lifetime. We also introduce a new metric named as PSTM (*patterned synchronization tendency metric*) to detect uneven traffic load distributions in the network due to the *patterned synchronization effect*. Based on the PSTM metric, our scheme dynamically adjusts the duty-cycle ratio of receivers in a self-organized manner not to have the duty-cycle patterns which could lead to the *patterned synchronization effect*. We evaluate our scheme against several representative cross-layer opportunistic forwarding schemes in the literature to show that our scheme significantly improve the network lifetime by detecting and reducing the occurrences of *patterned synchronization effect* in the network.

The rest of the paper is organized as follows. In Section II, some of related works are summarized. Section III introduces our proposed scheme in details. We also consider different other alternative approaches and their drawbacks as compared to our scheme. Section IV evaluates our scheme. Finally, some concluding remarks with future work are given in Section V.

II. Related Work

Opportunistic packet forwarding schemes have been widely applied in traditional MANETs due to their tolerance for highly mobile environments. Such techniques can also be applied in a WSN domain where sensors do not have mobility. Among existing works, we only consider the schemes proposed to reduce waiting delays of senders in duty-cycled MAC layer.

PaderMAC [2] is a strobed preamble MAC scheme which supports opportunistic routing and is built upon X-MAC [3]. It exploits the property that a dense network provides many routes to a destination. When forwarding a packet, a node starts sending short preambles containing the address of the destination. A neighbor, which has a route to the destination, can respond with a pre-acknowledgement to pick the data packet. Hence, the waiting delay of the sender can be reduced. In the presence of multiple potential receivers, a receiver contention mechanism is employed. However, it cannot avoid the *patterned synchronization effect* when a particular neighbor wakes up frequently and sends the pre-acknowledgement before other neighbors.

In [6], an opportunistic forwarding scheme (called ORW) for wireless sensor networks was presented. In contrast with PaderMAC, ORW completely runs on network layer assuming the availability of a suitable MAC protocol. To select a next hop neighbor for data transmission, ORW proposes a metric called EDC (*expected duty cycled wakeups*). Based on the EDC values of neighbors, each node creates its forwarder set and the first one to wake up from the forwarder set becomes the next hop. Even though ORW employs an approach to select a single forwarder, it still fails to prevent a particular neighbor from picking up the packet as a next hop too frequently. Therefore, the ORW protocol is also vulnerable to the *patterned synchronization effect*.

Among asynchronous MAC schemes, receiver-initiated MAC ones such as RI-MAC [7] are more effective since the first awaken neighbor can be easily identified by receiver beacons without wasting energy at the sender for preamble frames. Taking this advantage, the ORiNoCo data collection scheme for WSN is introduced in [8]. Nodes which generate data, forwards them towards the sink by using an opportunistically built routing tree based on a gradient variable. Each node can have many parents in the tree which provide a routing progress to the sink. A potential neighbor for the next hop should be a parent of the sender in the routing tree and should wake up when the sender is ready to transmit a packet. This behavior enables ORiNoCo to deliver data to the sink in a lower waiting delay at senders. ORiNoCo does not contain any mechanisms to identify the occurrence of *patterned synchronization effect*, while our proposed scheme provides a detection mechanism based on our proposed PSTM metric.

Opportunistic forwarding is not the only approach explored in the literature in order to reduce waiting delays. A geographical location based forwarding scheme named GeRaF was proposed in [10]. A sender transmits a frame containing its own and the destinations geographical location which are assumed to be known. If the frame is received by a neighbor located geographically closer to the sink than the sender, it will acknowledge the packet and forward it to another node using the same procedure. When there exist many neighbors available which are located closer to the sink than the sender, a receiver contention mechanism is applied. GeRaF always selects the next hop neighbor which is geographically located closest to the sink. Therefore, GeRaF is still vulnerable to the *patterned synchronization effect* when such a neighbor has a duty-cycle pattern similar to the sender.

III. Proposed Cross-layer Opportunistic Forwarding Scheme

A. Overview

We consider a WSN where a single sink is installed along with many static sensor nodes which are densely distributed and highly resource-constrained in terms of energy. Our proposed cross-layer opportunistic forwarding scheme is composed of a routing topology where each sensor node in the network has multiple partially-disjoint routes towards the sink node. That means each sensor node can have many upstream nodes which provide routing progress to the sink. Depending on the requirements of network lifespan and tolerable latency, a WSN should have an optimum duty-cycle ratio which is defined at the deployment time. In our scheme, each node try to maintain this predefined optimum duty-cycle ratio as much as possible. A MAC layer of the node performs a receiver-initiated medium access strategy where a node broadcasts a beacon whenever it wakes up from the duty-cycle. A data forwarder opportunistically chooses an upstream node among the neighbors which wakes up first, identified by a receiver-beacon of the neighbor.

To detect an occurrence of the *patterned synchronization effect*, nodes calculate the level of traffic load distribution based on our proposed PSTM metric and exchange it among neighbors by piggy-backing onto the packet headers. Receivers, identifying a situation of the *patterned synchronization effect*, slightly reduce their duty-cycle ratio to break the synchronization. Immediately after a node recognizes that the *patterned synchronization effect* is resolved, it switches back to the optimum duty-cycle ratio. This behavior helps the network to achieve a longer lifetime by maintaining the optimum duty-cycle ratio as much as possible while providing a lower

procedure CHECKPSTM
$PSTM_j \leftarrow calcPSTM()$ ▷ using Equation 2
if $PSTM_j <= PSTM_{thresh}$ **then**
$RDC_j \leftarrow RDC_{optimum}$
else if $PSTM_j > PSTM_{thresh}$ & $RDC_j > RDC_{min}$ **then**
$RDC_j \leftarrow (RDC_j - \Delta t)$
end if
end procedure

Fig. 1. Functionality of a receiver j in the network. $RDC_{optimum}$ and RDC_{min} values are defined at the deployment time based on the requirements of the network such as longer network lifetime and tolerable message latency. Δt represents an empirically defined value which will be deduced from the current duty-cycle ratio of the node to avoid a patterned synchronization effect.

latency due to the opportunistic strategy. Nodes work out of the optimum duty-cycle ratio only when they are resolving a *patterned synchronization effect*.

B. Detailed operations

In this section, the operation of our proposed scheme based on the PSTM metric is described in detail. It consists of two phases, namely *initialization* and *operational* phases. In the former phase, a routing topology is built to discover upstream nodes and the packet routing is performed in the latter phase with resolving *patterned synchronization effect* occurrences.

1) Initialization phase: Initially, all the nodes start operating in a non-duty-cycled state where all the radio transceivers are always switched on. The sink broadcasts a special frame which gets re-broadcasted by every receiver so that it floods throughout the network reaching all the nodes. Each intermediate node in the network may receive this special frame from many neighbors. Every such an intermediate node which receives a copy of the special frame, records the previous hop of the frame in its routing table as an upstream neighbor. In this way, all the nodes in the network (except the sink node) gets to know a set of upstream neighbors which can be used to deliver a packet towards the sink. Additionally, upon the reception of special frame, a receiver sends an acknowledgement to the sender so that each node can get to know their downstream neighbor details which is necessary for the PSTM metric calculation later. All the nodes in the network stay a predefined time period in the initialization phase in order to build the routing topology by identifying upstream and child nodes.

2) Operational phase: After the initial phase, all the nodes switch to the duty-cycled state where the predefined optimum duty-cycle ratio is maintained. All the nodes perform a receiver-initiated medium access strategy where a receiver beacon is transmitted each time a node wakes up from the sleep state. An awaken node waits for a predefined time frame for a potential packet reception. If it does not hear anything during the time frame, the node goes again to the sleep state without doing anything. In case of a collision, a receiver back-offs for a random period before sending a receiver-beacon again.

When a node has some data to send to the sink, it wakes up and waits for a receiver-beacon of an upstream neighbor. After hearing a receiver-beacon, this node forwards the packet to the receiver and waits for an acknowledgement. The receiver uses the same receiver-beacon as an acknowledgement to a packet reception. When there exists no acknowledgement received, a sender waits again for a receiver-beacon of an upstream neighbor and forwards the packet. The sender does not have to keep trying to forward the packet to a particular neighbor, since it may have many upstream neighbors which are suitable for the next hop. While working on the operational phase, both senders and receivers take actions to detect and resolve *patterned synchronization effect* occurrences, based on the PSTM which is described in detail as below.

C. Patterned Synchronization Tendency Metric (PSTM)

Each node maintains a special variable $transCount$ in the routing table per each of its upstream neighbor which gets incremented in each successful transmission. For example, a node i increments the variable $transCount_{j \leftarrow i}$ in its routing table when it successfully forwards a packet to its upstream node j. Thus, node i can keep track of the number of transmissions it has performed to each upstream neighbor. Additionally, in each successful packet transmission, a sender i calculates the probability of node i to select the particular upstream node j as the next hop for packet forwarding according to Equation 1. Then, it includes the resulting value $P_{j \leftarrow i}$ in the packet header.

$$P_{j \leftarrow i} = \frac{transCount_{j \leftarrow i}}{\Sigma\, transCount_{x \leftarrow i}} \quad (x \in N_i) \tag{1}$$

Upon reception of each packet, a receiver j gets to know the probabilities at which each of its child nodes have selected it by looking at the packet header. Here, $PSTM_j$ (*patterned synchronization tendency metric*) for node j is defined as the average of these probability values as shown in Equation 2. The $PSTM_j$ gets a higher value when the receiver j is getting selected as the next hop by its child nodes more frequently. The periodic operation of *patterned synchronization effect* detection at receivers, based on $PSTM$ is illustrated in the Figure 1. A node first calculates the $PSTM$. Then, it compares the result with a threshold value to identify a *patterned synchronization effect* situation. While the $PSTM$ is lower than the threshold, the node can keep running in the predefined optimum duty-cycle ratio. When $PSTM$ exceeds the threshold, the node should reduce its duty-cycle ratio by a Δt value. The values for the $PSTM$ threshold and Δt are decided empirically which are described in more detail in Section IV.

$$PSTM_j = \frac{\Sigma\, P_{j \leftarrow y}}{numChildNodes_j} \quad (y \in S_j) \tag{2}$$

Due to the reduction of duty-cycle ratio, a receiver can stay in the sleep state for a longer time, which leads to breaking the *patterned synchronization effect* of the receiver with its child nodes. For example, consider the network shown in Figure 2. Nodes A and B are upstream nodes of node S in the routing topology. Node B gets a patterned synchronization with node S which can be resolved by reducing the duty-cycle ratio of node B. To prevent the duty-cycle ratio from getting extremely lower values, a node can periodically subtract Δt from the duty-cycle ratio only if the network has not reached a minimum duty-cycle ratio. This minimum duty-cycle ratio should also be defined by the network operators, based on the tolerable message latencies caused by duty-cycles of nodes.

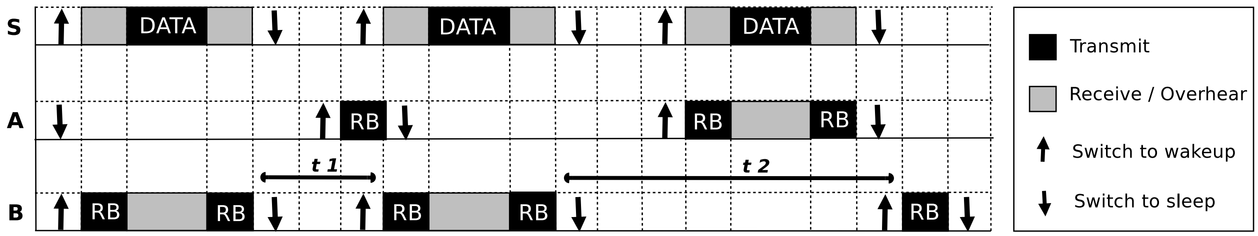

Fig. 2. A situation where a *patterned synchronization effect* occurs. Nodes A and B are upstream nodes of node S. When node S has a data packet to send, it wakes up and waits for a receiver beacon of an upstream node. Node B wakes up first and sends its receiver beacon so that the packet of node S is delivered to it. When node B successfully received the packet, it acknowledges the packet by sending another receiver beacon. After a while, node B detects a *patterned synchronization effect* occurrence and reduces its duty-cycle ratio to break the pattern. Later, it can reset its duty-cycle ratio back to the default value. ($t_1 < t_2$)

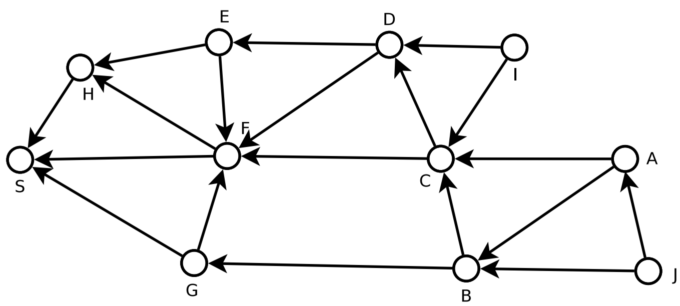

Fig. 3. A network in the operational phase. Arrow heads show the direction of upstream hops towards the sink.

TABLE I. ROUTING TABLE OF NODE A.

$NeighborID$	$Upstream$	$Downstream$	$transCount$
B	YES	NO	5
C	YES	NO	9
I	NO	NO	0
J	NO	YES	0

TABLE II. ROUTING TABLE OF NODE C.

$NeighborID$	$Upstream$	$Downstream$	$transCount$
A	NO	YES	0
B	NO	YES	0
F	YES	NO	24
D	YES	NO	13
I	NO	YES	0

D. Example scenario

Here, we introduce one example scenario to show how our proposed scheme works. Consider the network shown in Figure 3. After the initial phase, all the nodes know their upstream neighbors as well as the child nodes. Routing tables of nodes A and C are shown in the Table I and II. When node A has a packet to send to the sink, it wakes up and waits for a receiver beacon of an upstream neighbor either node C or B. Suppose node C wakes up first and transmits its receiver-beacon. Therefore, node A immediately sends the packet to node C and receives the acknowledgement. If a collision occurred at node C, it will go to sleep state without any acknowledgement. Then, node A has to wait again for a receiver-beacon from either node C or B to send the packet. According to the $transCount$ values in the routing table of node A, the value of $P_{C \leftarrow A}$ is 0.64 and it is embedded inside the packet header sent to node C. Suppose the values $P_{C \leftarrow I}$ and $P_{C \leftarrow B}$ are 0.53 and 0.72. Thus, $PSTM_C$ value becomes 0.63 on node C. If the acceptable level of load ($PSTM_{threshold}$) is set to 0.57, node C should reduce the duty-cycle rate accordingly since the $PSTM_C$ exceeds the threshold. Whenever a $PSTM_C$ calculation results in a value equal or lower than the threshold, node C will switch back to the default duty-cycle ratio of the network which is optimized for a longer network lifetime.

TABLE III. PARAMETERS OF THE SIMULATION SETUP.

Bandwidth	250Kbps
TX Range	250m
Carrier Sensing Range	550m
CCA Check Delay	128μs
SIFS	192μs
Distance Between Nodes	200m
Packet Size	28bytes

IV. PERFORMANCE EVALUATIONS

In order to investigate the performance of our proposed scheme, we compared it against two other well-known schemes; ORiNoCo[8] and PaderMAC [2] which are described in Section II. In addition, we evaluated it against an alternative third scheme which is a variation of our own scheme. In this scheme, after the initialization phase, all the nodes randomly adjust their duty-cycle ratio without detecting patterned synchronization effects unlike the proposed scheme.

Some performance metrics of interest were as follows.

1) network lifetime: the operational time period of the network from the starting point until the first node runs out of battery power. The network lifetime is directly affected by the patterned synchronization effect, causing packet forwarding nodes to run out of power too early if they keep being selected as next-hops.

2) waiting delay at a sender: the time a node waits holding a packet in a send buffer until it receives an opportunity to successfully transmit to a neighbor. This delay affects the power consumption of the sender which is in awaken state to forward the packet.

3) power consumption of the network: the average amount of power consumption in the network (defined as the total power usage divided by the total number of nodes).

A. Simulation setup

Using ns-2 simulator, we performed the evaluations of the proposed scheme against the representative schemes considering the selected performance metrics. We conducted performance comparisons using grid topologies where a sender and a receiver (the sink) are located in the two opposite corners of grid with various grid sizes (from 2×2 to 8×8 nodes). The sender transmits a stream of UDP packets towards the sink. Table III shows the parameters configured in our simulations.

In our proposed scheme, we have several system parameters, namely $RDC_{optimum}$, Δt and $PSTM_{thresh}$. $RDC_{optimum}$ highly depends on the network requirements as mentioned in a previous section. The optimum value should be determined based on the network operational lifetime, acceptable 1-hop packet delivery latency, energy consumption details of a node and etc. In our evaluations, we set the $RDC_{optimum}$ to 1% which is a commonly adapted value in the literature for duty-cycle MAC schemes. Δt should be as smaller as possible which helps to adjust the duty-cycle ratio smoothly. The adjustments we can make to the duty-cycle ratio of a node depends on the physical layer capabilities of a sensor node. For evaluations, we set Δt to 0.2%. $PSTM_{thresh}$ was selected empirically by running a lot of simulations to identify the point where the network achieves the highest lifetime. Figure 4 shows the results of measuring network lifetime against different $PSTM_{thresh}$ values. The network achieves the best lifetime when the $PSTM_{thresh}$ is set to 60%. Therefore we adopt this value as the $PSTM_{thresh}$ in further evaluations.

B. Simulation results

Figure 5 shows the variation of network lifetime in each of schemes with increasing network sizes. It is observed that our proposed scheme achieves a considerably higher network lifetime regardless of network sizes. While all the other cross-layer opportunistic forwarding schemes perform in a much similar way, our proposed scheme takes the advantage of the $PSTM$ metric to identify the occurrences of patterned synchronization effect. By adjusting the duty-cycle ratio, nodes are able to avoid the effect, leading to energy saving at the forwarders. This results in a longer network lifetime with an improvement of about 14%, as compared to other schemes.

The average waiting delay at the senders to forward a packet to the next hop is investigated. As shown in Figure 6, there is a slightly increased delay in our scheme. This is because our scheme tries to avoid a patterned synchronization effect by adjusting duty-cycle ratios. However, it does not deviate significantly from the other representative protocols, since the proposed scheme tries to maintain the optimum duty-cycle ratio as much as possible. In addition, as the number of neighbors increases, the proposed scheme becomes similar to other schemes in terms of sender waiting delay since the number of potential next-hop nodes for the sender increases. It implies that the proposed scheme is more suitable for highly dense networks where a sender has many neighbors which provide routing progress towards the sink.

Figure 7 shows the variation of the average power consumption. The average power consumption of our proposed scheme has no significant difference from the other alternative schemes. However, considering the maximum and minimum power consumption of the nodes in each of schemes, it is clear that the difference between those two values in the proposed scheme is lower as compared with the other schemes. This is because the load balancing functionality is achieved by avoiding the patterned synchronization effect in the network. Other schemes are unable to identify any occurrences of patterned synchronization effect and therefore some of their nodes may consume a lot of power to forward packets too frequently. In contrast, the proposed scheme smoothly distributes the packets from a sender among many next hop neighbors lowering the overhead on a particular node to consume too much energy for packet forwarding.

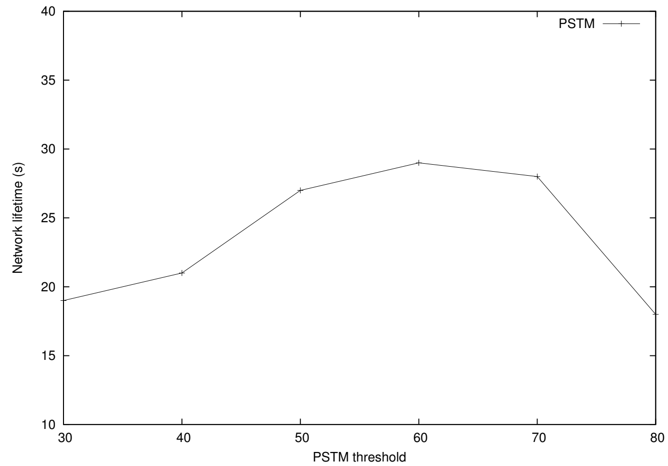

Fig. 4. The network lifetime with different $PSTM_{thresh}$ values. The network achieves the best lifetime when the value is set to 60%.

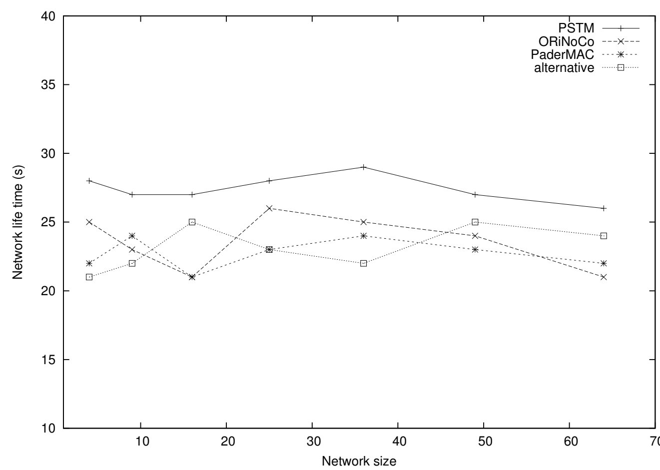

Fig. 5. Comparisons of network lifetime with different network sizes. Our proposed scheme achieves higher network lifetime.

As observed in evaluations, trying to dynamically adjust the duty-cycle ratio with random values can help to avoid the patterned synchronization effect. However, it does not accomplish the real purpose of solving the problem. The patterned synchronization effect causes a particular set of nodes in an opportunistic network to drain too much energy causing them to run out of battery. The purpose of an opportunistic network is to reduce the waiting delay at senders. When we randomly

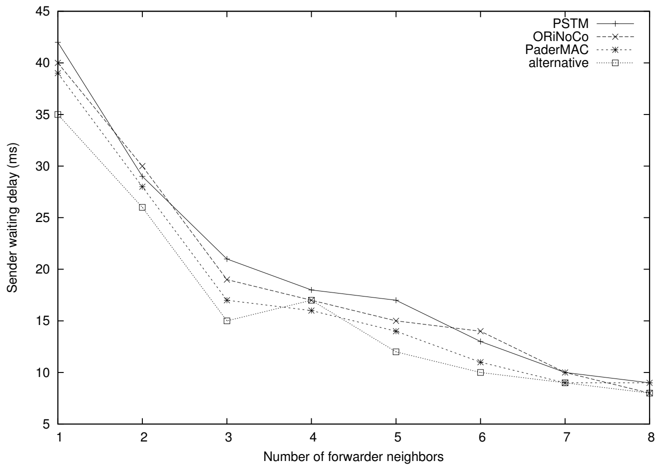

Fig. 6. The average waiting delay at a sender with increasing number of neighbors of the sender. Our proposed scheme does not introduce any overhead to the waiting delay as compared to other schemes.

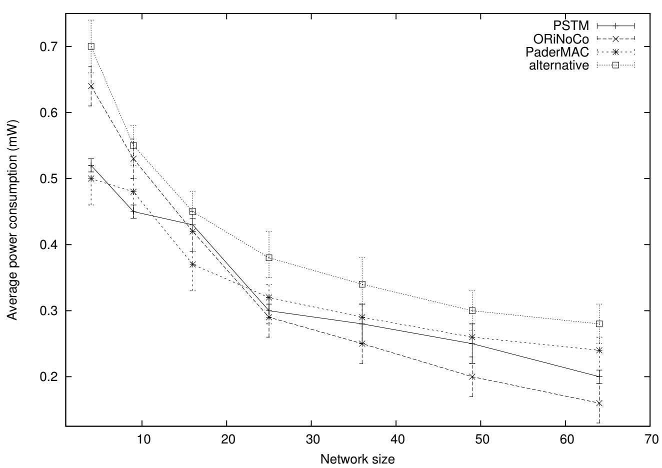

Fig. 7. The average power consumption of a node with the increasing network sizes. Additionally, the graph shows the highest and lowest power consumption values of the nodes in the network. The proposed scheme has a lower difference between the highest and lowest power consumptions of the nodes due to the reduction of the patterned synchronization effect occurrences.

adjust the duty-cycle ratio of the nodes, they become unable to maintain an optimum duty-cycle ratio since they may wake up too frequently at some points of time. Due to this reason, all the nodes in the network will spend energy more excessively than a network which maintains the optimum duty-cycle ratio. It shows the importance of maintaining an optimum duty-cycle ratio until a successful detection of a patterned synchronization as done in our scheme.

This work builds upon the fact that cross-layer opportunistic forwarding schemes reduces the waiting delay at the senders, which leads to higher energy efficiency. A recent survey presented in [5] clearly points out that load balancing is an important research problem in duty-cycled wireless sensor networks which has not been well studied. Our work deals with a similar kind of problem which is discussed in the survey solving a load balancing issue in a duty-cycled wireless sensor network.

V. Conclusion

In this paper, we proposed a new cross-layer opportunistic forwarding scheme for duty-cycled and highly resource constrained wireless sensor networks. Our scheme addresses the patterned synchronization effect which causes some nodes of the network to participate in packet forwarding too frequently than others. This situation makes such nodes to run out of their energy reducing the network lifetime. We introduced a new metric named as PSTM (*patterned synchronization tendency metric*) which is utilized to detect the occurrences of the effect. After the detection, our proposed scheme takes actions to resolve the effect by dynamically adjusting duty-cycle ratio of nodes in a self-organized manner. Extensive evaluations against different other representative schemes revealed that our scheme improves the network lifetime around 14% without delaying waiting time at senders, compared to other schemes. In our future work, we need to incorporate the proposed scheme with other metrics such as residual energy and link quality to make better decisions of next-hop forwarder. In addition, experiments using a real test-bed are planned to evaluate the proposed scheme.

Acknowledgment

This work was supported by the IT R&D program of MSIP/KEIT. [10041145, Self-Organizing Software platform (SoSp) for Welfare Devices].

References

[1] S. H. Ahmed, S. H. Bouk, A. Mehmood, N. Javaid, and S. Iwao. Effect of fast moving object on rssi in wsn: An experimental approach. In *Emerging Trends and Applications in Information Communication Technologies*, pages 43–51. Springer, 2012.

[2] M. Autenrieth and H. Frey. Padermac: a low-power, low-latency mac layer with opportunistic forwarding support for wireless sensor networks. In *Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW'11)*, pages 117–130. Springer, 2011.

[3] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys'06)*, pages 307–320. ACM, 2006.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, pages 1–14. ACM, 2009.

[5] K. Han, J. Luo, Y. Liu, and A. V. Vasilakos. Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. In *IEEE Communications Magazine*, pages 107–113. IEEE, 2013.

[6] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks (IPSN'12)*, pages 185–196. ACM, 2012.

[7] Y. Sun, O. Gurewitz, and D. B. Johnson. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys'08)*, pages 1–14. ACM, 2008.

[8] S. Unterschütz, C. Renner, and V. Turau. Opportunistic, receiver-initiated data-collection protocol. In *Proceedings of the 9th European conference on Wireless Sensor Networks (EWSN'12)*, pages 1–16. Springer-Verlag, 2012.

[9] O. Younis, M. Krunz, and S. Ramasubramanian. Node clustering in wireless sensor networks: recent developments and deployment challenges. *Network, IEEE*, 20(3):20–25, 2006.

[10] M. Zorzi and R. R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. *Mobile Computing, IEEE Transactions on*, 2(4):337–348, 2003.