

Multi-Tier Probabilistic Polling in Wireless Sensor Networks Powered by Energy Harvesting

Chisato Fujii¹ Winston K.G. Seah²

*School of Engineering and Computer Science
Victoria University of Wellington, New Zealand*

¹Chisato.Fujii@ecs.vuw.ac.nz

²Winston.Seah@ecs.vuw.ac.nz

Abstract—Wireless Sensor Networks (WSNs) that consist of battery powered sensor nodes start to lose operational capability once the nodes' batteries run out of power. To ensure the sustainability of WSNs, researchers have turned to alternative energy sources for power. Harvesting ambient energy from the environment to power WSNs is a promising approach but energy harvesting devices of the same footprint as wireless sensors are unable to provide sufficient energy for sustained operation. More likely, the energy harvested over a period is only enough for a sensor to sense and transmit a few packets before it needs to harvest more energy. Furthermore, the availability of energy varies depending on the environmental conditions and energy harvesting technology used. With sensor nodes operating in discontinuous periods, relaying data over multiple wireless links to remote data acquisition systems is a daunting challenge. In this paper, a multi-tier probabilistic polling scheme is proposed for multi-hop data delivery in wireless sensor networks that rely solely on energy harvesting for power. Unlike existing work which rely on simulations, we have implemented the scheme on commercial-off-the-shelf hardware and validated it experimentally to demonstrate its viability.

Keywords: Wireless Sensor Networks, Multi-hop data delivery, Energy Harvesting, Probabilistic Polling

I. INTRODUCTION

A Wireless Sensor Network (WSN) is typically made up of spatially distributed sensors that monitor the environment. These sensors transmit packets containing information such as temperature to the collection point, commonly referred to as a sink or an Access Point (AP). The sustainability of WSNs is purely dependent on the duration of sensor nodes' lifetime. The sensor nodes deployed outside the communication range of the sink rely on other sensor nodes to relay their packets towards the sink. When sensor nodes stop operating, the sensor network not only loses the sensing capability of those nodes but also other nodes that rely on these failed nodes to relay their sensed data to the sink. Aside from suffering physical damage or system failures, the most likely cause of a node's failure is the depletion of its battery, the predominant source of power for wireless sensors. Much of the research on wireless sensor networks has therefore focused on efficient methods to minimize energy usage in order to extend the lifetime of the nodes that form the network.

The concept of harvesting ambient energy from the environment and converting it into electricity to power sensor nodes can potentially lead to an unlimited power source for WSNs. While both WSN and energy harvesting (EH) are not new technologies, the utilization of EH in WSN remains challenging. Firstly, the EH device that is powering a wireless sensor node should be comparable in size but EH devices of the same footprint as a wireless sensor are unable to provide sufficient energy for sustained operation; more likely, the energy harvested over a period is only enough for a sensor to sense and transmit a few packets before it needs to harvest more energy. Secondly, the rate of energy harvesting is subjected to environmental conditions which introduces significant uncertainty in the availability of energy. Depending on the environment where the sensors are deployed, EH technology for WSNs are unlikely to provide a sustained supply to support continuous operation. They are likely to be able to provide only enough energy to power sensors sporadically and sensor nodes therefore need to exploit the sporadic availability of energy to quickly sense and transmit the data.

While there has been substantial research done on WSNs, almost all the existing research has focused on battery-powered WSNs [1]. New research on MAC protocols for WSNs powered solely by EH [2] has only started to appear very recently. In [3], various Carrier Sense Multiple Access (CSMA)-based and polling-based MAC protocols have been evaluated in terms of throughput and fairness using simulations that used harvesting rate data obtained from empirical characterization of commercial energy harvesting devices. The study also noted that sensor nodes waiting to synchronize in slotted MAC protocols is counter-productive as energy is consumed during the wait period and needs to be replenished with longer harvesting periods, thus leading to lower throughputs. Taking into consideration the constraints and salient features of EH-powered WSNs, a probabilistic polling protocol is then proposed [3]. The suitability of the Dynamic Framed-ALOHA (DFA) for use in an Energy-Constrained WSN with EH has also been studied [4]. Energy-on-demand provided by RF energy harvesting has initiated some studies on its effect on routing [5] and MAC protocol design [6]; however, the

ability to provide energy, albeit minute amounts, to sensor nodes when needed takes away the uncertainty faced by WSNs considered in [3] and [4].

In this paper, we extend the probabilistic polling protocol proposed in [3] for a single-hop WSN scenario to support multi-hop data delivery. In Section II, we present our multi-tier probabilistic polling (MTPP) protocol which enables the tier one (immediate neighbours of the sink) to poll nodes progressively further away. We then describe the experimental setup used to validate the MTPP protocol in Section III. To the best of our knowledge, all the MAC schemes proposed for use with EH-powered WSNs have only been validated using simulations, and this is the first testbed validation that is carried out using commercial-off-the-shelf devices. In Section IV, we discuss results of our functional tests carried out on the testbed which clearly show the daunting challenges that EH-powered WSNs currently face. We conclude with a discussion of ongoing and future work on MTPP and related areas.

II. MULTI-TIER PROBABILISTIC POLLING

Probabilistic polling has been proposed to achieve high throughput, fairness and scalability in EH-powered WSNs [3]. We extend this approach by enabling sensor nodes to poll other nodes beyond the range of the sink (as known as Access Point) to achieve a multi-hop WSN. In the single-hop probabilistic polling approach, the sink periodically broadcasts a polling message (Fig. 1) containing a probability value to all sensor nodes within its transmission range. Each sensor node generates a random probability value and compares it with the probability value it received from the sink. If the generated value is smaller than the received probability value, the sensor node transmits its data to the sink. The probability value is controlled at the the sink as shown in Fig. 2. If collision is detected by the sink, it decreases the probability value the next time it broadcasts the polling message. On the other hand, if the sink does not get response from any sensor node, it increases the probability value.

/*Probabilistic polling message format*/			
Tier	Probability		
0	1		
/* Data message format*/			
Tier	Hop Count	Data message	
0	1	2	3

Fig. 1. Message format

Since not all sensor nodes are deployed within a communication range of the sink, the sensor nodes outside the communication range have to relay the packet transmission via other sensor nodes. These intermediate nodes could either respond to the sink or poll further higher numbered tier sensor nodes outside the communication range of the sink.

Access Point
Initialise <i>probability</i>
Loop
Broadcast polling packet with <i>probability</i>
if no response
increase <i>probability</i>
else if collision was detected
decrease <i>probability</i>

Fig. 2. Pseudo-code for Access Point

In the following subsections, we describe how the probabilistic polling packet is passed to higher numbered tiers which are nodes further away from the sink. The multi-tier scheme is such that each sensor node belongs to the tier that corresponds to its distance from the sink represented by the number of intermediate hops.

A. Tier Initialisation and Transition

Each sensor is required to be initialised with a tier number when it joins the network. The tier number could be pre-assigned at deployment, but this may not fulfill the flexibility and scalability that could be achieved by dynamically assigning tier number depend on the given environmental condition. The interference caused by WiFi and other communication devices is likely to cause fluctuations in link quality and vary the number of intermediate hops between any node and the sink.

Prior to joining the network, sensor nodes are initialized with a tier number of "0xff" denoting the highest tier number. If it receives any broadcast polling message which has a lower tier number than its own, it assigns its tier to the received tier number plus one. Sensor nodes check if the received broadcast polling message contains a tier number lower than its parent hierarchy (i.e. tier number that is one lower than its current tier) and it immediately updates its tier number if it notices a much lower tier number in the polling message.

Conversely, if a node stops receiving broadcast (polling) messages from lower numbered tiers but it is able to receive broadcast messages from sensor nodes in the same or higher numbered tiers, then it updates its tier to the received tier number plus one. This occurs in scenarios where the communication range has decreased due to surrounding environmental changes. The detection of this is done using a counter that gets incremented each time the sensor node receives polling messages from tiers that are not lower than its current tier. Once the counter reaches the threshold value, the node updates its tier number to the received tier number plus one. We set the threshold value to ten, indicating that the sensor node transits to a higher numbered tier after it received 10 consecutive broadcast messages from tiers that are not its parent. When a sensor receives a broadcast message from a lower numbered tier at any time, the counter is reset. A value of 10 provides enough time for the sensor node to ensure that the interference level is consistently high enough that the sensor node was no longer in its parent tier's communication range. This approach effectively deals with dynamic environments where

the surrounding conditions and ambient interference vary.

B. Tier-independent Polling Approach

A simple way to implement multi-tier polling is by the sink polling tier one nodes and then a polled sensor node broadcasts the same probabilistic value to the higher numbered tier and waits until it receives the data, or a preset timer expires; this is similar to a blocked remote procedure call in distributed systems. Another way to implement multi-tier polling is to enable each tier to independently poll higher numbered tier nodes once a nodes receives probability value, as shown in Fig. 3. The *Message* data stores its measured data as well as messages received from higher numbered tiers in the format specified in Fig. 2. The received data from higher numbered tier sensor nodes are added to end of the *Message* data. The *HopCount* field indicates the number of hops that a certain data packet has been relayed by sensor nodes. With the probability value generated that was lower than the received polling value, the sensor node either transmits the data or polls higher tier sensor nodes. The decision is made by *turnTPollLowerTier* which keeps track on the action in the previous duty cycle so that the polling of higher numbered tier and the data transmission were done one after another. This scheme takes into consideration the fact that transmission and reception of the packets are energy expensive and multiple transactions in one duty cycle should be minimized. Furthermore, in EH-powered WSNs, waiting consumes energy too and should be avoided.

End Device
<pre> Message data = new Message{HopCount = 1, tier = this.tier} Message lowerTierData boolean turnToPollLowerTier = true Loop RandomN ← Random value (between 0 and 1) Receive polling packet from this.tier-1 containing <i>probability</i> if RandomN < <i>probability</i> if turnToPollLowerTier is false if Transmit data is successful lowerTierData ← empty data.HopCount = 1 turnToPollLowerTier ← true else if lowerTierData is empty Broadcast <i>probability</i> to lower tier lowerTierData ← received <i>Message</i> data.HopCount ← lowerTierData.HopCount+1 add lowerTierData to the end of data turnToPollLowerTier ← false Sleep and harvest energy </pre>

Fig. 3. Pseudo-code for End Device

A node in a tier should initiate its own polling cycle based on the probability in the polling message it receives that grants it the permission to transmit or not. This means that while the sink polls tier-one sensor nodes, the polled tier-one sensor node either polls tier-two sensor nodes with the same probability or simply transmits its data back to the sink. Each sensor node then stores the data it receives from the higher numbered tier sensor nodes that it has polled, until it gets

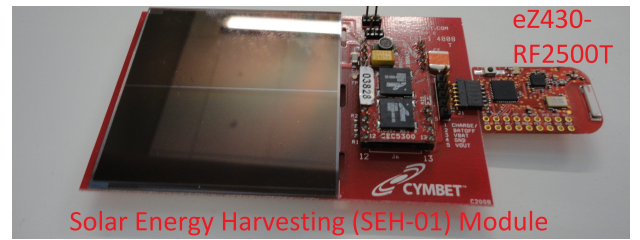


Fig. 4. End Device (SEH and eZ430-RF2500T)

polled by lower numbered tier sensor nodes. Once a node gets polled, it transmits a data message (Fig. 1) containing its own data and other data it received from higher tier nodes that it previously polled. The data will be relayed from tier to tier until they eventually reach the sink.

In this approach, it is effective that when a node gets polled, it contains the data from its higher tier and thus it can respond to the polling message instantaneously without waiting for the message from higher numbered tiers. The duration of the period where a node is in the listening state is shortened thus it could minimise the use of energy.

C. Broadcast-based Transmission Approach

All communication between the sensors is accomplished using link layer broadcast. Although using the broadcast approach produces more collisions, it consumes less energy than the destination specific transmission since it avoids unnecessary link layer coordination procedures. Furthermore, collisions are minimized by the ordered access provided by the probabilistic polling approach.

D. Link Layer Collision Response

Collisions cannot be entirely avoided even with the probabilistic polling approach and adjusting the probability value in the polling message can alleviate the level of collisions. Sensor nodes respond to collision in the following manner. Each sensor node stores the data and listens to the channel to determine if its transmission succeeded, exploiting the implicit acknowledgment feature provided by the wireless media. If its transmission was unsuccessful, the sensor node waits until it gets polled again to retransmit the data.

III. EXPERIMENTAL SETUP

Texas Instruments (TI) provides a platform called the MSP430 Solar Energy Harvesting (SEH) Development Tool [7]. It contains the thin film battery storage EnerChip on SEH which is used to store the energy and when the light source is too weak to provide operational energy. This was used for the experiment hardware platform and sensor nodes were deployed in a indoor environment.

The SEH uses the low-power RF network protocol SimpliciTI [8]. This provides API for the transmission of packets between End Devices and the Access Point. The TI eZ430-RF2500T transceiver [9] was used for the communication between the Access Point and the End Devices, as shown in Fig. 4. The Access Point is the collection point of information,

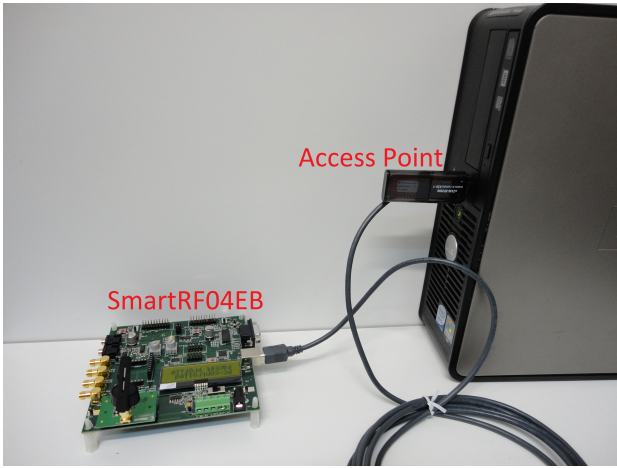


Fig. 5. Access Point and SmartRF04EB

and it connects to a workstation via USB port as shown in Fig. 5. The eZ430-RF2500T transceiver is embedded inside the USB device. The End Devices are sensor nodes comprising a Solar Energy Harvester and the transceiver as shown in Fig. 4. They are spatially distributed for collecting data and transmitting them to the Access Point.

SimpliciTI is partially designed for the energy harvesting aware system; however, the provided API does not include the efficient packet scheduling required in ambient energy powered End Devices. If an End Device is deployed beyond the communication range of the Access Point, it requires an intermediate node, called a Range Extender, which forwards data towards the Access Point. This Range Extender requires battery power since it is not designed to sleep. Using these Range Extenders in the energy-harvesting environment degrades the sustainability of WSNs since Range Extenders' lifetime limits the whole WSN's lifetime. A WSN which consists only of ambient energy powered sensor nodes enhances the sustainability. Thus, permanently powered devices (Range Extenders) are replaced with ambient powered End Devices, which are now required to perform the forwarding functionality as well as collecting and transmitting their own data.

The eZ430-RF2500T transceivers are programmed in C using the Code Composer Studio platform [10] for compiling and installing the program. The multi-tiered probabilistic polling algorithm defined in Section II was implemented using the SimpliciTI API on the transceivers. This eZ430-RF2500T transceiver requires minimum of 3V for the operational energy. Once the voltage drops below this threshold, the transceiver has to switch to the harvesting mode. All other functionalities are turned off for the duration of harvesting period and its duration depends on the deployed environment. The device gets woken up once it charges up to 3.6V.

Ten End Devices were deployed arbitrarily in an indoor environment. Fig. 6 shows the topology used in the two-tier WSN. Each End Device initializes its tier by listening for the lower numbered tier polling messages. Once it gets

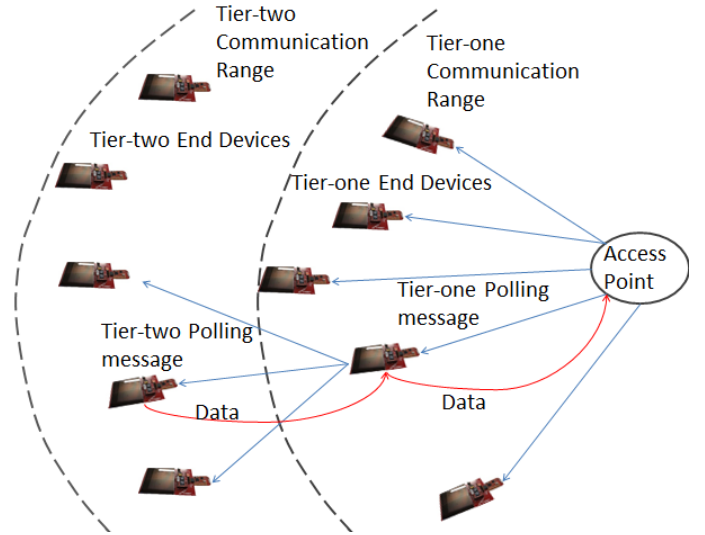


Fig. 6. The Multi-tier WSNs topology

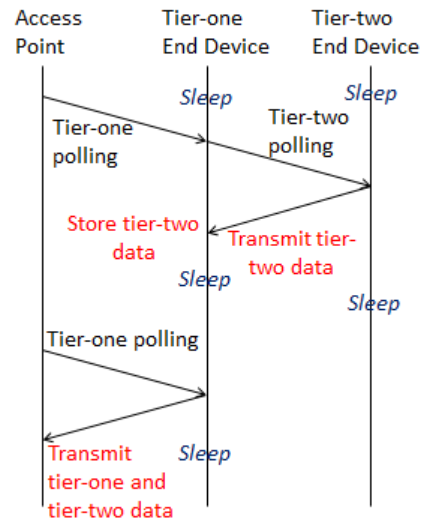


Fig. 7. Polling and data message exchange

polled by the lower numbered tier, it either polls the higher numbered tier or transmits its own data message in each turn. Fig. 7 shows the exchange of polling and data messages in the experiment. The data content from tier-two was stored at a tier-one End Device for one duty cycle until it receives the next polling packet after harvesting enough energy. The SmartRF Studio tool [11] was used to interpret the captured packet from SmartRF04EB sniffer [12], which was located at the Access Point as shown in Fig. 5. The End Devices' harvesting behaviour as well as the duty cycles were monitored using Tektronix Digital Oscilloscope [13].

IV. FUNCTIONAL TEST RESULTS

The messages in Fig. 8 were captured using the SmartRF04EB sniffer and interpreted with SmartRF Studio. The first eleven bytes were used for system reserved parameters. The important content in the polling message is the tier

number and the polling probability value corresponding to the last two bytes of the polling message that are highlighted. The polling message sent from the Access Point to tier-one End Device contains the tier number of 0x00 and polling value of 0xfc. This is the highest possible probability value and thus each node is guaranteed to transmit the data as long as it receives the polling message. This was appropriate for this experiment since there were only ten End Devices deployed. As the number of End Devices increases, it would be necessary to reduce the polling probability value as to reduce the contention among devices as their number increases.

```

/* probabilistic polling from the Access Point*/
16:39:04.750 | ff ff ff ff 36 94 04 6c 3f 24 6a 00 fc |
                                     tier probability

/* data message from the tier-one End Device*/
16:39:04.780 | ff ff ff ff 5e 27 d4 44 3f 63 23
01 01 57 0a 00 00 00 00 |
tier-one data

/* probabilistic polling from the tier-one End Device */
16:42:09.890 | ff ff ff ff 5e 27 d4 44 3f 24 98 01 fc |
                                     tier probability

/* data message from the tier-two End Device*/
16:42:09.947 | ff ff ff ff 74 d9 9b c3 3f a3 d5
02 01 00 0a 00 00 00 00 |
tier-two data

/* probabilistic polling from Access Point*/
16:42:31.703 | ff ff ff ff 36 94 04 6c 3f 24 9e 00 fc |
                                     tier probability

/* data message from tier-one End Device
also containing tier-two data*/
16:42:31.721 | ff ff ff ff 5e 27 d4 44 3f 63 35
01 02 04 cc 02 01 00 0a |
tier-one data tier-two data

/* Probability value decremented from 0xfc to 0xfb*/
16:42:31.787 | ff ff ff ff 36 94 04 6c 3f 24 a0 00 fb |
                                     tier probability

```

Fig. 8. Captured Messages

The tier number in the polling message is either 0x00 indicating the message is from the Access Point, or 0x01 from tier-one End Devices. The polling message follows the structure defined in Fig. 1 which uses the last two bytes. The data message format is also shown in Fig. 1. The first byte of the highlighted tier-one data indicates the tier number of the End Device, and the second byte is used to indicate *Hop Count*; the number of data contained which includes its own data and data from higher numbered tiers. When a tier-one End Device contains stored tier-two data, it is placed into last four bytes of the data content as shown in Fig. 8 and second byte *Hop Count* defined in Fig. 1 was incremented.

The data message transmitted from the tier-two End Device to the tier-one End Device was also stored in the first four bytes of the data content as highlighted in Fig. 8. It gets processed at the tier-one End Device and the data content from the tier-two device was shifted up to the last four bytes.

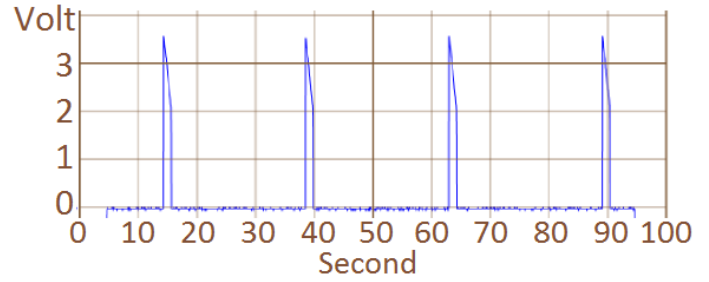


Fig. 9. Duty Cycle of the End Device

Furthermore, the *Hop Count* of the tier-one data content was incremented indicating that it contains two data sets. This field was used to determine the successful function of two tier transmission.

A. Duty Cycle and the Illuminance

A critical factor of probabilistic polling is that each End Device must be able to receive the polling packets when they are active. With the deployment in the indoor environment, the fully solar powered nodes were able to become active after a period of between 20 and 30 seconds. The minimum illuminance for the End Device to operate is 200 lux [14]. An End Device deployed in the environment below this threshold is continuously non-operational. However, as soon as the illuminance increases above this threshold the End Device powers up and becomes functional. The variation of illuminances, even in the same room depending on the deployed location, made the duration of the active period of the End Devices significantly different. Each End Device experienced differences in their illuminance parameters, including illuminance variation during daytime and nighttime, slight differences in deployed location and instantaneous coverage from the luminous source due to other objects. These factors affected the duty cycle of energy harvesting and thus the duty cycle was unique for each End Device. Fig. 9 shows the End Device duty cycle in the arbitrary indoor environment. From this figure, the peak voltage was measured as 3.56V. The duration of active state was 1.3 seconds and inactive state of 23.2 seconds. This gives a duty cycle of 0.056, which makes the operational duration significantly lower than the inactive duration.

B. Receiver Radio Control

The receiver (RX) radio must be turned on to enable reception of packets. The RX at the Access Point was kept always on, thus packets were received at all times. The content in the RX buffer was retrieved using a *SimpliciTI* API call. Since contents were temporarily stored in RX, it was not critical to retrieve contents frequently from the RX as long as the RX buffer was not full. On the other hand, End Devices cannot keep their RX radio on all the time due to energy consumption of the RX radio. Thus, the idea was to keep the RX radio active for a short period of time, but still long enough to be able to complete the remaining tasks that must be executed in each duty cycle.

Although the duration of the RX radio active state is short at End Devices, End Devices must be able receive the probabilistic polling packet from the Access Point while the RX radio turned on. End Devices did not have enough energy to keep RX active for longer than 33ms after executing other required tasks. To ensure that End Devices receive the polling packet within this period, a fixed probabilistic polling interval 33ms was used.

C. Probabilistic Polling Rate

Since the polling rate from the Access Point is very fast, by the time the Access Point receives the data packet from an End Device, it has already transmitted another polling packet. This becomes an issue if the data packet transmitted from a polling cycle and another data packet from consecutive polling cycle happen to arrive at the Access Point at the same time.

However, the advantage of using probabilistic polling is that the possible difference with the probability values in consecutive polling packets can be kept within one percent. Thus, it would not become an issue unless the data processing time at End Devices is excessively large compared to polling rate. In this case, the received probability value may be outdated. By assuming that the RTT is much smaller than the polling rate and also the processing time at each End Device is similar, the outdated polling could still effectively used.

D. Probabilistic Polling Control

The probabilistic polling value has been implemented to be incremented when there was no response from any End Device. The possible range of probability values range from zero to one, and increments of 0.0025 was used. However, following experimentation with this incremental value, it has been modified to 0.01 due to the fact that the deployed number of sensors was relatively small, ranging from ten to twenty at a time, and this meant that each End Device almost always had a successful transmission without any collision. Thus, the probability value was mostly close to one, and at the start up of the system, using the incremental value of 0.01 enabled the system to stabilize faster. This optimal value is expected to vary for WSNs with more End Devices being deployed. The probabilistic polling protocol is expected to be more effective in WSNs with over 200 sensor nodes [3]. With regard to the time interval needed to increase the probability value across a 0.01 range, there is possibly an optimal rate. If this increase was too fast or too slow, the performance of probabilistic polling may be adversely effected and fail to stabilize.

E. Dynamic Tier Assignment Performance

Within the deployed environment, the interference from other devices such as WiFi varied over time. During hours where the strong interference was present, some of the tier one End Devices were pushed to tier two because they were unable to listen to the Access Point's polling messages. The implementation of automatic adjustment between tiers has been effectively achieved by End Devices listening for polling message in every duty cycle to determine which tier they

belonged to. End Devices always listened for the polling messages and updated their tier number based on the lowest numbered tier that they have heard from.

V. CONCLUSION

We have proposed a multi-tier probabilistic polling (MTPP) that enables sensor nodes in a multi-hop WSN to dynamically configure their tier position depending on the polling messages received. Similar to the basic probabilistic polling approach, the polling probability value is generated and controlled by the sink. The sink then polls the first tier nodes with this probability value and each sensor node that received the polling message will either transmit its data back to the sink or re-broadcast the polling message to the higher tiers. The process is repeated and data from higher tiers are progressively relayed back to the sink via intermediate tiers. We have implemented the MTPP protocol on commercially available devices and conducted experiments to validate its viability. While we have only reported results showing a two-tier testbed, there is significant potential to scale the protocol to larger networks despite the daunting challenges faced. This is also the first reported study of a MAC protocol designed for WSNs powered solely by energy harvesting that has been validated by real experiments using a proof-of-concept prototype.

REFERENCES

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [2] W. K. G. Seah, Z. A. Eu, and H.-P. Tan, "Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) – Survey and Challenges," in *Proceedings of the First International Conference on Wireless Communications, Vehicular Technology, Information Technology and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Aalborg, Denmark, 17-20 May 2009.
- [3] Z. A. Eu, H.-P. Tan, and W. K. Seah, "Design and performance analysis of mac schemes for wireless sensor networks powered by ambient energy harvesting," *Ad Hoc Networks*, vol. 9, no. 3, pp. 300–323, 2011.
- [4] F. Iannello, O. Simeone, and U. Spagnolini, "Dynamic framed-aloha for energy-constrained wireless sensor networks with energy harvesting," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, Miami, FL, USA, 6-10 Dec 2010.
- [5] R. Doost, K. Chowdhury, and M. Di Felice, "Routing and link layer protocol design for sensor networks with wireless energy transfer," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, Miami, FL, USA, 6-10 Dec 2010, pp. 1–5.
- [6] J. Kim and J.-W. Lee, "Energy adaptive mac protocol for wireless sensor networks with rf energy transfer," in *Proceedings of the Third International Conference on Ubiquitous and Future Networks (ICUFN)*, Dalian, China, 15-17 June 2011, pp. 89–94.
- [7] "MSP430 Solar Energy Harvesting Development Tool (eZ430-RF2500-SEH) from Texas Instruments." [Online]. Available: <http://www.it.com>
- [8] "SimpliciTI Compliant Protocol Stack from Texas Instruments." [Online]. Available: <http://www.it.com>
- [9] "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver from Texas Instruments." [Online]. Available: <http://www.it.com>
- [10] "Code Composer Studio (CCStudio) Integrated Development Environment (IDE) from Texas Instrument." [Online]. Available: <http://www.it.com>
- [11] "CC2550/CC2550DK Development Kit from Texas Instruments." [Online]. Available: <http://www.it.com>
- [12] "SmartRF Studio tool from Texas Instruments." [Online]. Available: <http://www.it.com>
- [13] "TDS2022B from Tektronix." [Online]. Available: <http://www.tek.com>
- [14] "eZ430-RF2500-SEH Solar Energy Harvesting Development Tool User's Guide from Texas Instruments." [Online]. Available: <http://www.ti.com/lit/ug/slau273c/slau273c.pdf>