

Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks

Jaemin Jeong, David Culler
EECS Department, UC Berkeley
Berkeley, CA 94720, USA
{jaemin,culler}@eecs.berkeley.edu

Jae-Hyuk Oh
United Technologies Research Center
East Hartford, CT 06108, USA
OhJ@utrc.utc.com

Abstract—Previously, many researchers sought to optimize the throughput and power consumption by radio transmission power control, but the experimental platforms and the workloads did not reflect the reality of WSNs. We present a dynamic transmission-power-control algorithm based on previous studies and evaluate the algorithm using realistic multi-hop WSN workloads and a large Mica2dot-based testbed. We have found that potential gains of dynamic transmission-power control are much smaller than what is indicated by prior research. Compared to the fixed transmission-power control, the dynamic transmission-power control improves the power consumption up to 16% for convergence traffic, but no noticeable performance improvements for aggregation traffic. The effect of dynamic transmission-power control becomes larger as we reduce the radio duty cycle, with 37% power savings at a 10% duty cycle. This result suggests that dynamic transmission-power control is most useful in combination of low-power MAC protocols which implement radio low duty cycling.

Keywords—Transmission Power Control, Traffic Pattern, Duty Cycling, Neighbor Selection Method

I. INTRODUCTION

Wireless sensor networks (WSNs) are becoming possible choices for supporting data communication in industrial applications such as HVAC and building control that have been traditionally based on wired links. For these industrial applications, it is required that a sensor network support multi-hop routing. This is needed to deliver data traffic over a long distance when nodes do not have direct line of sight. In addition, a high data rate is often desirable to deliver the data with a throughput that is comparable to the data rate of wired links. In order to achieve a higher data rate, one may increase the data sending rate of each sensor node. However, it has been observed in previous works [1]–[3] that increasing the data rate does not directly translate to a proportionate increase in the throughput; this is due to increased contention among the nodes.

Previously, a number of researchers have tried to achieve the best possible throughput of power consumption of the network by adjusting the radio transmission power of individual nodes (dynamic transmission-power control). However, these previous works have some limitations to be applied to multi-hop WSNs. First, some of them are based on idealized simulators such as ns-2 [2], OMNet++ [4], OPNET [1] and Rooftop toolkit [3]. Second, the previous works do not reflect various

traffics in multi-hop routing. Third, some of the previous works have used high power consuming hardware platforms that do not reflect the power consumption profile of WSNs (e.g. WaveLAN radio [5], PC104 nodes [6]). Fourth, some of the previous works have projected energy consumption only for transmission state not considering the overall radio energy consumption including sleep or idle listening state [6], [7].

In this paper, we study the possible improvements and limitations of dynamic transmission-power control on multi-hop WSNs in the following steps: First, we present a generalized dynamic transmission-power-control algorithm based on various transmission-power-control algorithms previously reported by researchers. Second, we consider various traffic patterns in multi-hop WSNs as workloads for dynamic transmission-power-control algorithm. Third, we evaluate the algorithm on a large Mica2dot-based WSN testbed.

The main finding from increasing the level of realism and fidelity and using traffic patterns that occur most often in actual WSN deployments in this study of transmission-power control is a rather pessimistic one. The potential gains are much smaller than what is indicated by prior enthusiastic results. Our detailed findings are as follows: first, for divergence traffic, a fixed transmission-power control strategy is sufficient to achieve the best possible throughput. Second, on convergence traffic, dynamic transmission-power control consumes about 16% less power than fixed transmission-power control. Third, for aggregation traffic, dynamic transmission-power control does not have performance improvements over fixed transmission-power control. Fourth, the power savings due to dynamic transmission-power control do become larger as we reduce the radio duty cycle. Our mathematical model shows that dynamic transmission-power control consumes about 37% less power than fixed transmission-power control when we employ 10% radio duty cycle for convergence traffic. This result suggests that dynamic transmission-power control is most useful in combination with low-power MAC protocols that implement radio low duty cycling.

The rest of this paper is organized as follows: We review the related works in Section II. We describe our proposed algorithm and the various traffic patterns for WSNs in Section III-A and Section III-B. We present the experimental methodology in Section IV and experimental results in Section V and

we conclude this paper in Section VI.

II. RELATED WORKS

There are previous studies that sought to optimize performance by varying the radio transmission power. The basic ideas of these dynamic transmission-power-control algorithms are similar to one another. The algorithms find how many neighbors each node has and adjust the radio transmission power of each node so that the number of neighbors stays within desired range. LINT/LILT [3], LMA/LMN [4], PCBL [6] and ATPC [7] are such examples. These algorithms vary depending on the following parameters: (1) how neighbors are chosen, (2) what metric is optimized for, (3) how the experiment is set up and what traffic patterns are considered.

As for the neighbor selection method, three different methods have been used in the literature: connectivity based [4], packet-reception-rate (PRR) based [6] and received-signal-strength (RSS) based [1], [3], [7]. In LINT/LILT, a node maintains a list of neighbors whose RSS values are higher than the threshold, and it adjusts the radio transmission power if number of neighbors is outside the predetermined bound. In LMA/LMN, a node determines its range by counting how many other nodes acknowledged to the beacon message it has sent. In the algorithm of ElBatt *et al.* [1], each node ranks neighboring nodes in the order of RSS and adjusts the radio transmission power so that it covers only N-most neighbors. In PCBL, each sensor node adjusts its radio transmission power for each destination node with the smallest possible value such that $PRR > PRR_{threshold}$, and it also filters out the nodes that have too low PRR values. In ATPC, each sensor node maintains the link quality information (RSSI or LQI) for each neighbor node and adjusts the radio transmission power individually for each neighbor.

These previous works seek to improve either throughput or power consumption. From the experimental results they provide, we can make the following observations:

First, using transmission-power control does not improve throughput over fixed transmission power and its maximum possible throughput is no higher than the best case with fixed transmission power. LINT/LILT [3] had maximum throughput at the density of 0.5 nodes/sq.miles and this throughput is no higher than that of fixed transmission-power control (NONE). ElBatt *et al.* [1] showed that varying transmission power for each radio link produced best possible throughput of 0.365 at connectivity range N of 4 while setting transmission power the same for all links produced maximum throughput of 0.355 at N = 9. PCBL (dynamic transmission-power control algorithm) [6] had the best possible packet delivery rate of 98.7% and this is very close to packet delivery rate of 99.2% for M-BL (maximum transmission power with blacklisting). ATPC [7] achieved 98% of packet reception rate and this is no better than 100% of packet reception rate for maximum transmission power. Second, the previous works showed that radio transmission-power control reduced power consumption. LMA/LMN [4] had about 37,000 and 42,000 seconds for the

TABLE I
DIFFERENT TRANSMISSION-POWER-CONTROL ALGORITHMS

Ramanathan <i>et al.</i>: (LINT/LILT)	
Metrics	Throughput, transmission power, delay
Neighbors	Received-signal-strength based
Experiment	Simulation (Rooftop C++ Toolkit)
Traffic	Point-to-point routing between two random points
Kubisch <i>et al.</i>: (LMA/LMN)	
Metrics	Network lifetime, connectivity
Neighbors	Connectivity based
Experiment	Simulation (OMNet++)
Traffic	Point-to-point routing between two nodes
ElBatt <i>et al.</i>:	
Metrics	Throughput, transmission power
Neighbors	Received-signal-strength based
Experiment	Simulation (OPNET)
Traffic	Random point-to-point routing
Son <i>et al.</i>: (PCBL / M-BL)	
Metrics	Packet reception rate (Throughput)
Neighbors	Packet-reception-rate based
Experiment	PC 104 nodes with Mica2 as radio transceiver
Traffic	Point-to-point routing using Directed Diffusion
Lin <i>et al.</i>: (ATPC)	
Metrics	Packet reception rate, transmission power
Neighbors	RSSI used, LQI shown to work
Experiment	Outdoor experiment (MicaZ nodes, CC2420 radio)
Traffic	Multi-hop convergence traffic

network lifetime (= inverse of power consumption) compared to 26,000 seconds of fixed transmission-power control (42% and 62% improvement in the network lifetime). ElBatt *et al.* showed that setting the radio transmission power differently for different next-hop nodes consumed about half the power compared to setting the radio transmission power the same level for all next-hop nodes (15mW vs. 33mW at connectivity range N = 4). PCBL [6] consumed 53.2% less energy for broadcasting than M-BL. ATPC [7] consumed power around 53.6% of the maximum transmission power.

Although these previous studies claimed possible improvements in throughput and power consumption, they have limitations to be applied to multi-hop wireless sensor networks. First, some of the previous works were demonstrated with idealized simulators [1], [3], [4]. Second, these previous works did not consider various traffic pattern patterns in multi-hop wireless sensor networks [1], [3], [4], [6], [7]. Third, some of the previous works were tested with high power consuming hardware platforms that had different power consumption profile from wireless sensor networks [5], [6]. Fourth, some of the previous works projected energy consumption only for transmission state not considering the overall radio energy consumption including sleep or idle listening state [6], [7]. Since the radio channel of a sensor node is in sleep or idle listening state much of time, their projection of energy consumption can be quite different from actual mote energy consumption.

There are IEEE 802.11 protocol specific optimization techniques such as adjusting transmission power of RTS/CTS control packets [2], [8], [9], changing transmission power based on packet size or fragmenting a packet into smaller fragments [5]. Since fixed small sized packets are usually used in WSNs, these techniques put severe overhead or do not fit into the structure of WSNs.

In this paper, we will present a generalized form of dy-

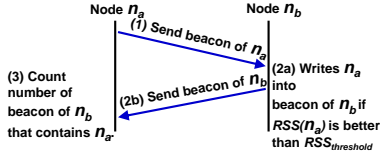


Fig. 1. Finding Number of Effective Neighbors

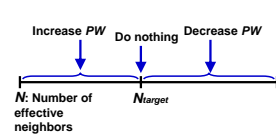


Fig. 2. Adjusting Radio Transmission Power

dynamic transmission-power-control algorithm that is based on previous studies on transmission-power control (Section III-A). Then, we will evaluate the possible improvements by the algorithm in terms of throughput and power consumption in the context of realistic multi-hop routing workloads (Section III-B) and WSN testbed platform (Section IV).

III. DESIGN

A. Description of Transmission Power Control Algorithm

The radio transceiver of a sensor node consumes current at different rates depending on its state: sleep, idle listening or transmission, and setting the radio transmission power changes the current consumption at transmission state. Since increasing the radio transmission power has both positive (reduced number of hops to the destination) and negative effects (increased interference among nodes), the radio transmission power needs to be set to the right level to achieve the best performance. The radio transmission power of each node can be set to a fixed value (fixed transmission-power control: FTPC), or it can be adjusted dynamically so that each node has similar number of neighbors (dynamic transmission-power control: DTPC).

Based on the previous studies on transmission-power control, we present a generalized form of dynamic transmission-power-control (DTPC) algorithm that attempts to improve the throughput and power consumption of multi-hop WSNs. The algorithm works in two steps:

- 1) Probe the messages from other nodes and count how many neighbors a sensor node has.
- 2) Adjust the radio transmission power of the sensor node until it hears desired number of neighbors.

In order to adjust the radio transmission power to the right level, we use the notion of an effective neighbor. A node n_e is an effective neighbor of n_a if n_a knows that n_e can receive beacon messages from n_a . A sensor node can tell its number of effective neighbors (N) using the following protocol (Figure 1).

- 1) Each node n_a sends a beacon message.
- 2) A node n_e that hears the beacon message from n_a with the link quality better than a predefined threshold $RSS_{threshold}$ records the source ID of the message. When the node sends its beacon message, it piggybacks the list of neighbors on the beacon message.
- 3) Node n_a hears a beacon message from n_e and it can tell whether n_e has heard n_a by looking at the neighbor

list which is piggybacked on the beacon message. Node n_a counts all such n_e 's that have heard n_a .

After finding the number of neighbors, our algorithm adjusts the radio transmission power so that number of effective neighbors N converges to a predefined value N_{target} (Figure 2).

As a way of choosing a neighbor, we have considered two methods: a connectivity-based method and a RSS-based method that can be easily implemented in our platform with a single instantaneous measurement of connectivity or RSS. To compare these two methods, we measured the average throughput of the sensor nodes (the experimental platform is explained in Section IV) for the traffic where data from each node converges to the destination node. We observed that the RSS-based method had 0.348 packets/sec and the connectivity-based method had 0.311 packets/sec as average throughput when each node sent messages at a rate of 0.5 packets/sec. This result shows that that the RSS-based method is more effective than the connectivity-based method, and we have used the RSS-based method as our neighbor selection method for our further experiments. We have also found that the RSS-based method can behave differently depending on the $RSS_{threshold}$ value that filters neighbors out of all the nodes a node has heard. For example, we observed that the average throughput for the RSS-based neighbor selection method was 0.350 packets/sec for $RSS_{th} = 100$ and 0.375 packets/sec for $RSS_{th} = 50$.

The dynamic transmission-power-control (DTPC) algorithm initializes the transmission power P , the step D and the sign S as follows:

$$P_0 = P_{max}, D_0 = D_{init}, S_0 = 1$$

At step 1, the algorithm either increases or decreases the transmission power P_1 depending on whether the number of neighbors N is larger or smaller than N_{target} :

$$\begin{aligned} S_1 &= \text{sign}(N_1 - N_{target}) \\ D_1 &= D_{init} \\ P_1 &= \min\{P_0 - S_1 D_1, P_{max}\} \end{aligned}$$

where $\text{sign}(x)$ is a function which is defined as follows:

$$\text{sign}(x) = 1 \ (x \geq 0) \ \text{or} \ -1 \ (x < 0)$$

At step $k \geq 2$, the algorithm either increases or decreases the transmission power as it did at step 1. In addition, the algorithm divides the step D_k by 2 if the sign S_k is different from the previous step S_{k-1} . This algorithm makes the step D_k smaller as N_k moves close to N_{target} so that the radio transmission power converges to a desired value:

$$\begin{aligned} S_k &= \text{sign}(N_k - N_{target}) \\ D_k &= \max\left\{\frac{3 + S_k S_{k-1}}{4} D_{k-1}, 1\right\} \\ P_k &= \max\{\min\{P_{k-1} - S_k D_k, P_{max}\}, P_{min}\} \end{aligned}$$

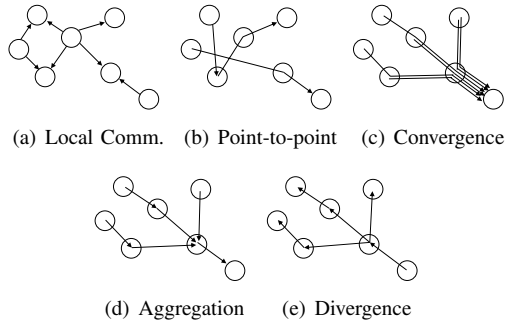


Fig. 3. Different Traffic Patterns

B. Traffic Patterns

Different sensor-network applications can cause a wide variety of traffic patterns. The traffic of WSNs can be either single hop or multi hop. Multi-hop traffic patterns can be further divided depending on the number of sender and receiver nodes, or whether the network supports in-network processing. Based on these criteria we can categorize the traffic patterns into (a) local communication, (b) point-to-point routing, (c) convergence, (d) aggregation and (e) divergence. Figure 3 illustrates different traffic patterns.

Local communication is used to broadcast the status of a node to its neighbors and is also used to transmit the data between the two nodes directly. Point-to-point routing is used to send a data packet from an arbitrary node to another arbitrary node, and this is commonly used in wireless LAN environment. Point-to-point routing is not widely used for WSNs because building a routing tree does not scale well for sensor nodes that have only small data memory. With convergence traffic pattern, the data packets of multiple nodes are routed to a single base node. Convergence traffic pattern is commonly used for data collection in WSNs. With aggregation traffic pattern, the data packets can be processed in the relaying nodes and the aggregate value is routed to the base node rather than the raw data. Finally, the divergence traffic is used to send a command from the base node to other sensor nodes.

Previous works for dynamic transmission-power control (DTPC) did not consider different traffic patterns in WSNs because they were based on wireless LAN where point-to-point routing or local communication is more common [1], [3], or they just used the traffic patterns that could be readily supported by their experimental platforms [4], [6].

As a preliminary experiment, we measured the throughput of fixed transmission-power control for convergence, aggregation and divergence traffic patterns using the platform explained in Section IV. For convergence and aggregation traffic patterns, each node sent its own data packets at a rate of 1 packet per 2 seconds and the throughput was calculated by counting the packets that successfully arrived in the base node for each source node. For the divergence traffic pattern, the base node sent broadcast messages at a rate of 1 packet per 2 seconds. We measured the throughput for each non-base

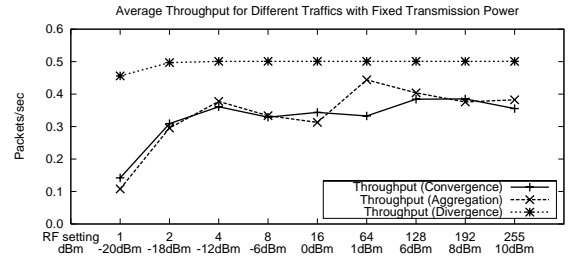


Fig. 4. Throughput for Different Traffic Patterns (Fixed Transmission Power)

node by counting the report messages that are sent through the UART port.

From this experiment, we have observed the following results (Figure 4): As we increased the radio transmission power, the average throughput of convergence and aggregation traffic patterns gradually increased and reached the peak fluctuating around 0.35 packets/sec. For divergence traffic, the average throughput quickly increased and reached around 0.5 packets/sec. Considering that the best possible throughput for each node is 0.5 packets/sec, this result implies that convergence and aggregation traffic patterns still need improvement while the divergence traffic does not. Thus, in the rest of this paper, we cover only convergence and aggregation traffic patterns for further experiments.

IV. EXPERIMENTAL METHODOLOGY

A. Experimental Platform

In order to monitor the wireless communication behavior of a real sensor network, we have used a subset of 22 nodes in the Smote testbed in the Computer Science building at UC Berkeley [10] that consists of total 78 Mica2dot sensor nodes. Each sensor node can be programmed and monitored through the Ethernet programming board connected to the sensor node. Smote provides a convenient test environment by exposing the UART serial port of each mote as a TCP port. A testbed user can easily monitor the behavior of the overall sensor network as well as each individual sensor node by hearing the messages from the TCP connections. A Mica2dot sensor node has the Chipcon CC1000 radio transceiver [11], which allows a user to change the radio transmission-power output between -20dBm to 10dBm (corresponding to current consumption between 5.3mA and 26.7mA) by setting the radio transmission power from 1 to 255. Although we have used Mica2dot-based testbed as our experiment platform, the dynamic transmission power control can be applied to other WSN platforms with adjustable radio transmission power and radio-provided link quality information (e.g. Telos, MicaZ, Mica2 or XSM).

As for the software components of a mote application, we have used TinyOS B-MAC protocol [12] (without low-power listening) and TinyOS *MintRoute* multi-hop routing protocol [13]. By default, *MintRoute* supports convergence traffic. By monitoring the messages that arrived in the base node through the *Receive* interface, we can measure the end-to-end throughput for each node in the network. *MintRoute*

also exposes the *Intercept* and *Snoop* interfaces, and this allows monitoring of neighborhood information (e.g. number of neighbors) without affecting the function of the routing module. Although *MintRoute* itself does not support aggregation traffic, we can easily modify *MintRoute* to support aggregation traffic by intercepting the forwarded messages and piggybacking the messages into the sensor node's own message.

B. Performance Metrics

For the experiments, we measure the following performance metrics: (1) end-to-end data throughput, (2) power consumption, (3) neighbor distribution and routing information.

1) *Throughput*: The end-to-end throughput for node i is defined as the rate of number of data packets from node i that have arrived at the base node ($nPackets$) over the elapsed time since the first packet from node i has arrived ($Duration$). In order to measure the end-to-end throughput for each node, we have built a host application that counts the traffic and calculates the throughput for each node through the connected base node.

2) *Power consumption*: The average power consumption of node i can be attributed to the following factors: radio state, duty cycling, transmission power and message sending rate:

- Radio state: A sensor node draws different levels of current depending on its radio state: sleep (I_{sleep}), idle listening (I_{listen}) or transmission (I_{tx}).
- Duty cycling: The average current consumption of a sensor node changes depending on duty cycling. With α -duty-cycling ($0 \leq \alpha \leq 1$), the current draw of a node can be formulated as follows:

$$\alpha \cdot I_{listen} + \frac{T_{tx}}{T_{period}} \cdot (I_{tx} - I_{listen}) + (1 - \alpha) \cdot I_{sleep} \quad (1)$$

- Transmission power: The average current consumption of a sensor node can change depending on the level of transmission power (RF_{tx}).
- Message transmission rate: As we see in the above formula, the current draw of a sensor node depends on the message transmission rate ($1/T_{period}$). In multi-hop routing situation, the number of messages a node transmits is also dependent on the number of children nodes (N_{child}). A child node of node n_a is a node from which node n_a forwards packets in a *MintRoute* routing tree.

We have measured the average current consumption rate of a Mica2dot node using a Fluke-189 data-logging multimeter while varying transmission power (RF_{tx}) and number of children nodes (N_{child}). For example, Figure 5 shows the trend of current consumption for $N_{child} = 0$ and $RF_{tx} = 192$. From this measurement, we have built a mapping table between the experiment parameter tuple (N_{child}, RF_{tx}) and the average current consumption as shown in Figure 6. Based on this mapping table, we can estimate the current consumption for each node in the routing tree.

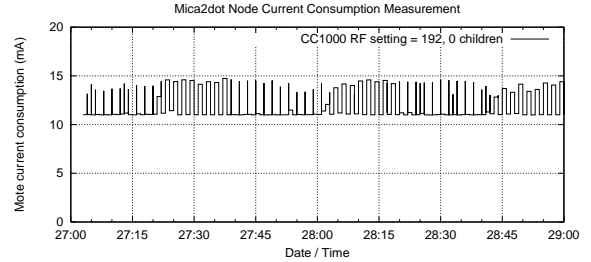


Fig. 5. Trend of current consumption of Mica2dot with RF transmitter power level 192 and no children nodes being forwarded. The current consumption of the sensor node was averaged over each sampling period (1 Hz)

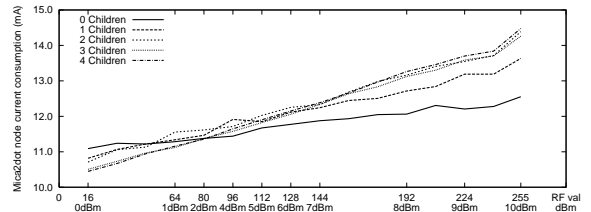


Fig. 6. Average current consumption of Mica2dot node for different configurations of RF transmission power and number of children nodes (N_{child}) being forwarded

3) *Neighbor distribution and routing information*: A neighbor of a sensor node is a node that is just being heard or heard with RSS higher than a certain threshold. Neighbor distribution shows how the algorithm adapts to an optimum state when it is given different initial parameters. Routing information like number of children nodes (N_{child}) shows how much traffic will increase due to the routing. In order to monitor the neighbor distribution and routing information, we have built a host application that logs the information of each node through the TCP port allocated to each sensor node in the Smote testbed.

V. EXPERIMENTAL RESULTS

For the experiment, we measure the performance of the dynamic transmission-power control (DTPC) comparing it with that of the fixed transmission-power control (FTPC) for convergence and aggregation traffic patterns. Table II summarizes the experiment setting.

A. Convergence Traffic Results

1) *Overall behavior*: For the fixed transmission-power control (FTPC), the power consumption increases as we increase the radio transmission power RF_{tx} from 64 through 255. At the same time, the throughput also increases accordingly (Figure 9).

The dynamic transmission-power control (DTPC) has a little bit different trend (Figure 9). The power consumption increases as we increase the number of target neighbors N_{target} from 3 through 15. But, compared to FTDC, a smaller number of nodes actually increase their transmission power. The throughput has a trend that increases up to $N_{target} = 12$ and decreases after that point as we further increase N_{target} .

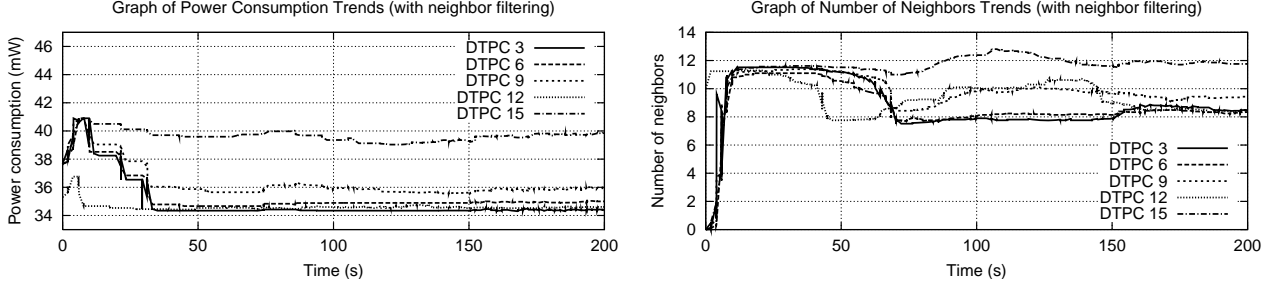


Fig. 7. Trends of Power Consumption and Number of Neighbors of DTPC for Convergence Traffic Averaged over All Nodes

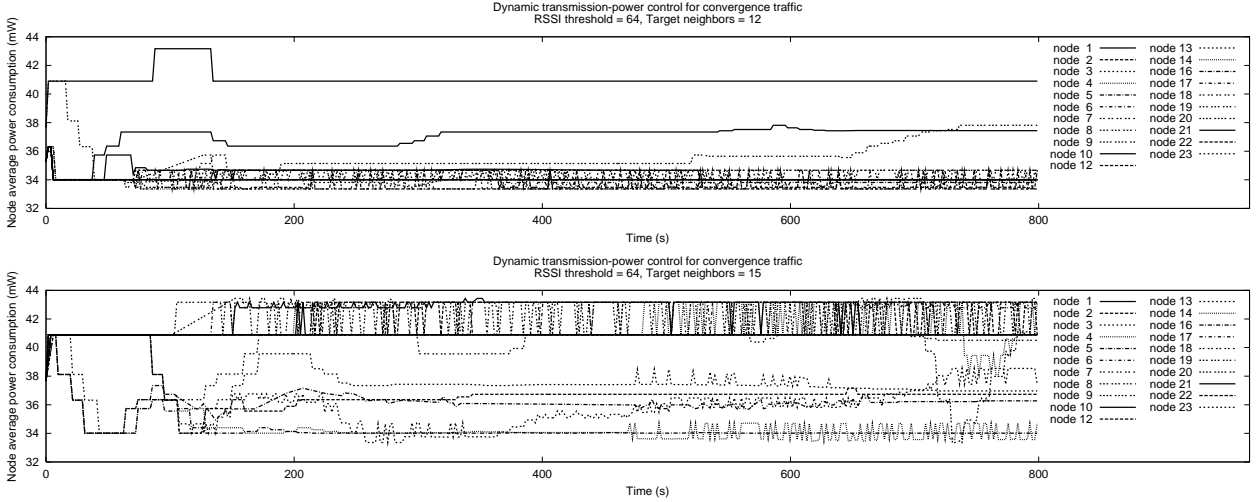


Fig. 8. Trends of Power Consumption for DTPC. (a) DTPC-12 ($N_{target} = 12$), (b) DTPC-15 ($N_{target} = 15$)

TABLE II
EXPERIMENT SETTING

Parameter	Configuration
Measurement Time	20 min
Transmission power update period	11 sec
Sending Rate ($1/T_{period}$)	1 packet per 2 sec
Only for FTPC	
Initial transmission-power value RF_{tx}	64, 128, 192, 255
Only for DTPC	
Number of target neighbors N_{target}	3, 6, 9, 12, 15
Threshold for received signal strength $RSS_{threshold}$	50

2) *Temporal Behavior*: Figure 7 shows the trends of power consumption and the number of neighbors for DTPC. We can see that the power consumption and the number of neighbors stabilize after 3 minutes (= 180 seconds).

The temporal behavior of DTPC for individual nodes (Figure 8) shows the importance of choosing the best experiment parameters for better performance. We logged the temporal behavior of DTPC for different configurations ($N_{target} = 3, 6, 9, 12, 15$) and compare the two contrasting cases.

As for DTPC-12 ($N_{target} = 12$), the transmission power of each node became stable in less than 3 minutes and only two nodes changed their transmission power during the rest of the experiment. Whereas, for DTPC-15 ($N_{target} = 15$), the transmission power of sensor nodes settled in longer time

(5 minutes) at initial stage. Even after the initial settlement, more number of nodes (7 nodes) adjusted their transmission-power levels. The change in the transmission-power level is reflected in the node current consumption trend. We can see that DTPC-15 has higher temporal variation than DTPC-12. The more frequent change in transmission-power level causes more radio interference and this explains why DTPC-12 has higher throughput than DTPC-15.

3) *Comparing FTPC and DTPC*: Figure 10 compares the performance at the level of individual nodes for the two cases: FTPC at $RF_{tx} = 255$ and DTPC at $N_{target} = 12$. The performance of the two cases can be summarized as follows:

- FTPC: average throughput 0.390 packets/sec, power consumption 41.34mW at $RF_{tx} = 255$
- DTPC: average throughput 0.409 packets/sec, power consumption 34.63mW at $N_{target} = 12$

DTPC achieves 4.87% higher throughput and 16.23% less power consumption than FTPC when both algorithms achieve the maximum throughput. The graph of the number of neighbors (Figure 10(c)) gives a clue why DTPC consumes less power. Sensor nodes, especially high ranked ones, have more neighbors with FTPC than with DTPC. Having more neighbors results in more traffic in the routing tree and higher power consumption.

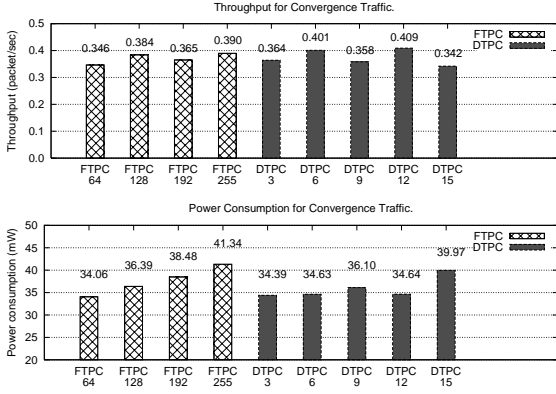


Fig. 9. Comparing FTPC and DTPC for **Convergence Traffic**: (1) Throughput, (2) Power Consumption

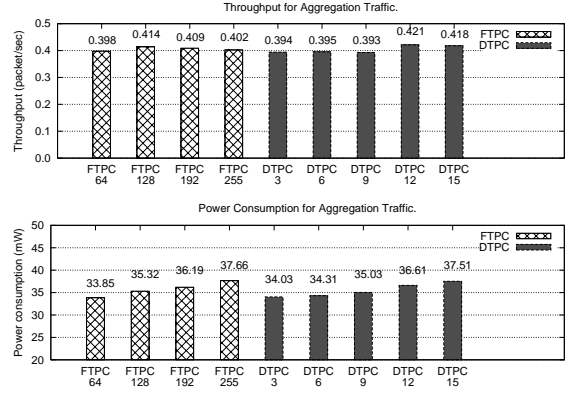


Fig. 11. Comparing FTPC and DTPC for **Aggregation Traffic**: (1) Throughput, (2) Power Consumption

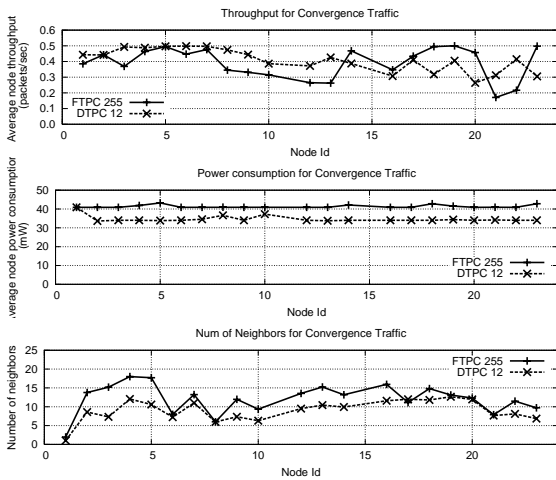


Fig. 10. Comparing FTPC-255 ($R_{F_{tx}} = 255$) and DTPC-12 ($N_{target} = 12$) for **Convergence Traffic**: (1) Throughput, (2) Power Consumption, (3) Number of Neighbors

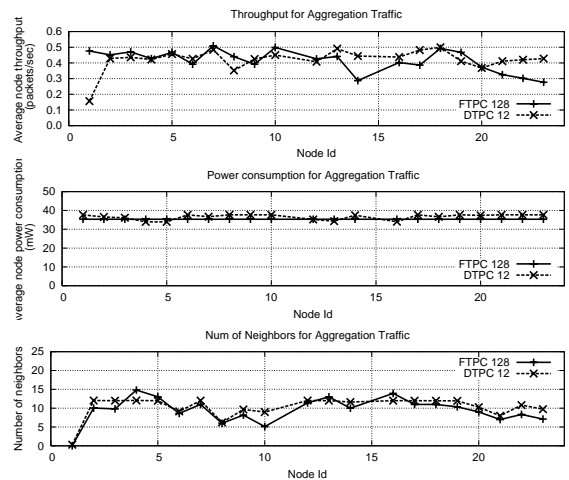


Fig. 12. Comparing FTPC-128 ($R_{F_{tx}} = 128$) and DTPC-12 ($N_{target} = 12$) for **Aggregation Traffic**: (1) Throughput, (2) Power Consumption, (3) Number of Neighbors

B. Aggregation Traffic Results

1) *Overall Behavior*: As we can see from Figure 11, the throughput of both FTPC and DTPC vary in small amount as we increase the radio transmission power. And the power consumption increases as we increase the transmission power (FTPC) or the number of target neighbors (DTPC).

2) *Comparing FTPC and DTPC*: Figure 12 compares the throughput and the power consumption of the two algorithms for the case where each algorithm has best throughput: $R_{F_{tx}} = 128$ for FTPC and $N_{target} = 12$ for DTPC. We can see that the two algorithms have about the same level of throughput overall. As for the power consumption by radio transmission, FTPC consumes smaller power than DTPC. This is because sensor nodes have fewer neighbors with FTPC than DTPC (Figure 12(c)). The performance of the two algorithms can be summarized as follows:

- FTPC: average throughput 0.414 packets/sec, power consumption 35.32mW at $R_{F_{tx}} = 128$
- DTPC: average throughput 0.421 packets/sec, power consumption 36.61mW at $N_{target} = 12$

We can see that DTPC improves the throughput by 1.69% while it consumes 3.65% more power for the radio transmission than FTPC. For aggregation traffic, DTPC does not have performance improvement compared to FTPC.

3) *Comparing the Performance for Aggregation Traffic with the Performance of Convergence Traffic*: Figure 13 shows the traffic reduction by using aggregation traffic patterns. Using aggregation traffic reduced the traffic that goes into the base node by 40.5% to 57.5% out of all the traffic originating from each node.

C. Estimating the Effect of Duty Cycling

From earlier this section, we have seen that the effect of adjusting radio transmission power is not so large compared to the manufacturer-provided power consumption of the radio transceiver. This is because the sensor nodes spent small time in transmission state and they stayed in idle listening state without using radio duty cycling.

Assume that each node duty cycles its radio while sending messages at a rate of 1 message per 2 seconds. Then, we

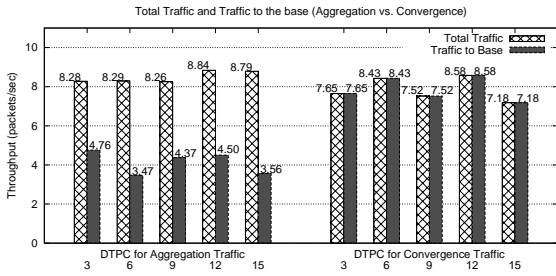


Fig. 13. Aggregate Traffic to the Base Node

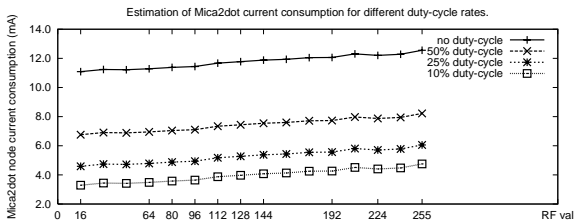


Fig. 14. Estimation of Mica2dot node current consumption for different duty-cycle rates with no children nodes in the routing tree

can estimate the average current draw of a Mica2dot node for different duty cycle rates by using Equation (1) with empirical measurement of $I_{sleep} = 1.848mA$ and $I_{listen} = 10.514mA$. Figure 14 shows the estimated average current draw of a Mica2dot for the case the sensor nodes duty cycle at 50%, 25% or 10% with convergence traffic and no children nodes in its routing tree.

Given the average current draw at different duty-cycle rates, we can estimate the power consumption for different configurations of transmission-power-control algorithms. Table III shows the power consumption estimate for FTDC with $RF_{tx} = 255$ and DTDC with $N_{target} = 12$. We can see that DTDC becomes more effective as we reduce the duty-cycle rate. With 50% duty-cycle rate, DTDC has 23.64% less power consumption than FTDC, and with 10% duty-cycle rate, DTDC consumes 37.34% less power. The power consumption estimates for different configurations of FTDC and DTDC are shown in Figure 15.

Although this result is based on mathematical modeling, a number of low-power MAC protocols such as B-MAC [12] and S-MAC [14] have already implemented the radio low duty cycling. Since the radio duty cycling can be done independently of the radio transmission-power control, this estimation result shows that we can improve the power saving of WSNs by using both low-power MAC and radio transmission-power control.

VI. CONCLUSION

In this paper, we have evaluated the performance of the dynamic transmission-power control (DTDC) using a Mica2dot-based WSN testbed and compared it with that of the fixed transmission-power control (FTDC). We have tested the algo-

TABLE III
ESTIMATION OF POWER CONSUMPTION FOR FTDC AND DTDC WITH DUTY CYCLING

Duty-Cycle Rate	FTDC-255	DTDC-12	Improvement
No duty cycle	41.34mW	34.64mW	16.21%
50% duty cycle	28.34mW	21.64mW	23.64%
25% duty cycle	21.84mW	15.14mW	30.68%
10% duty cycle	17.94mW	11.24mW	37.34%

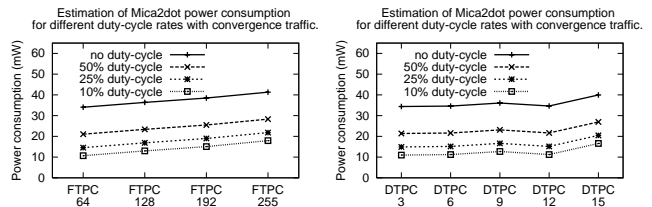


Fig. 15. Estimation of Power Consumption for FTDC and DTDC at Different Duty-Cycle Rates

gorithms with popular sensor-network traffic patterns. From the experiments, we have found the following: For traffic patterns that *diverge* from a common source, using FTDC is sufficient to achieve the best possible throughput. For scenarios where the traffic patterns *converge* to the source, we have found that DTDC reduces about 16% of power consumption over FTDC while it makes little improvement in throughput. But, for aggregation traffic pattern, DTDC does not have any significant performance improvement over FTDC. The savings in power consumption become larger when we employ duty cycling. For convergence traffic with 10% duty cycling, DTDC can consume about 37% less power than FTDC. This result suggests that dynamic transmission-power control is most useful in combination with low-power MAC protocols that implement radio low duty cycling.

REFERENCES

- [1] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao, "Power management for throughput enhancement in wireless ad-hoc networks," *IEEE ICC*, 2000.
- [2] J. P. Monks, V. Bharghavan, and W.-M. W. Hwu, "A power controlled multiple access for wireless packet networks," *IEEE INFOCOM*, 2001.
- [3] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," *IEEE INFOCOM*, 2000.
- [4] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabae, "Distributed algorithms for transmission power control in wireless sensor networks," *IEEE WCNS*, 2003.
- [5] J.-P. Ebert, B. Stremmel, E. Wiederhold, and A. Wolisz, "An energy-efficient power control approach for wans," *Journal of Communications and Networks (JCN)*, September 2000.
- [6] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," *IEEE SECON*, October 2004.
- [7] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "Atpc: adaptive transmission power control for wireless sensor networks," *ACM Sensys*, 2006.
- [8] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu, "Intelligent medium access for mobile ad hoc networks with busy tones and power control," *IEEE Journal on Selected Areas on Communications*, September 2003.
- [9] E.-S. Jung and N. H. Vaidya, "A power control mac protocol for ad hoc networks," *ACM MOBICOM*, September 2002.
- [10] "Smote testbed," <http://smote.cs.berkeley.edu>.
- [11] "Cc1000 single chip very low power rf transceiver data sheet," http://www.chipcon.com/files/CC1000_Data_Sheet_2.2.pdf.
- [12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," *ACM Sensys*, November 2004.
- [13] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," *ACM SENSYS*, November 2003.
- [14] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," *IEEE INFOCOM*, 2002.