Transmission power control based on packet reception rate

Conference Paper · March 2014

DOI: 10.1109/NTMS.2014.6814057

4 authors, including:

Mariusz Slabicki
Nokia Solutions And Networks, Wroclaw, Poland

Tomasz Surmacz
Wroclaw University of Science and Technology

Bartosz Wojciechowski
Wroclaw University of Science and Technology

Transmission Power Control Based on Packet Reception Rate

Maciej Nikodem, Mariusz Słabicki, Tomasz Surmacz, Bartosz Wojciechowski
Wrocław University of Technology, Institute of Computer Engineering, Control and Robotics
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Email: {maciej.nikodem; mariusz.slabicki; tomasz.surmacz; bartosz.wojciechowski}@pwr.wroc.pl

***Abstract*—This paper proposes and analyses properties of transmission power control mechanism for Wireless Sensor Networks based on estimated packet reception rate. The algorithm is dedicated for low power wireless networks and its aim is to minimise average cost of radio transmission, reduce retransmissions and interferences between network nodes.**

I. Introduction and related work

Wireless sensor networks (WSNs) are composed of large number of elements (nodes) that are self-sufficient, operate independently but coordinate their actions with other nodes in the same network. In most applications nodes have individual, limited power resources, are equipped with short range, low power radio transceivers, microcontroller, and various sensors (e.g. temperature, humidity, CO_2, CO, etc.) [1], [2]. Many consider limited power supply, unattended operation and maximization of network lifetime as crucial aspects of WSN design [1]. Network lifetime can be prolonged through duty cycling and optimised radio transmissions – data aggregation, energy-efficient routing, low-power medium access control protocols, and Transmission Power Control (TPC) [1], [3], [4].

TPC serves two purposes. First, it allows to reduce power consumption by adjusting transmission power $\mathrm{P_{Tx}}$ to the lowest level sufficient for effective communication. Then, it reduces radio congestion and assures desired network topology and connectivity. In most cases the two goals are not contradictory. Empirical analysis by Jeong et al. [5] confirms that combined with low-power duty-cycling MAC protocols, TPC algorithms allow for power savings of over 30%.

Various approaches to TPC may be employed in varying use-cases, depending e.g. communication channel properties. Algorithms based on connectivity adjust $\mathrm{P_{Tx}}$ based on the number of required neighbours [6]. Other algorithms adjust transmission power dynamically for communication with each neighbour based on acknowledgement (ACK) counts of successful and failed transmissions [7]. If properties of communication channel do not change often, it is beneficial to adjust $\mathrm{P_{Tx}}$, based on channel or link quality estimation. This estimation can be done either by calculating the Packet Reception Rate (PRR) [8] or measuring and averaging the Received Signal Strength Indicator (RSSI) [3]. Local algorithms base their decisions only on information from the node's immediate vicinity, hence may lead to globally suboptimal decisions, but with the advantage of low overhead and very good scalability, which is an important aspect for such highly distributed systems.

II. Transmission power control algorithms

Existing TPC algorithms can be generalised into a basic structure composed of 4 stages: packet transmission, communication statistics update, failed transmission handling, and successful transmission handling. *Packet transmission* includes channel access procedures (e.g. clear channel assessment), transmission of packets, and waiting for their acknowledgements (ACKs). Lack of ACK in a predefined time indicates transmission failure. *Communication statistics update* includes any procedures that are executed by the transceiver to gain information about communication channel, transmitter and receiver. This may include measurements of channel noise, interferences, channel attenuation or calculation of packet error/reception rate, bit error rate, etc. For the transmitting node the easiest way to verify whether the transmission succeeded is based on ACK. Some additional actions may be taken in *successful transmission handling*, but are usually optional. *Failed transmissions handling* is more complex – transmission may fail either because $\mathrm{P_{Tx}}$ is too low or temporary communication conditions were bad. In the latter case, the transmitter should simply repeat the transmission while in the former – also increase the $\mathrm{P_{Tx}}$. If $\mathrm{P_{Tx}}$ cannot be increased any further and transmissions still fail, then the message is dropped. In such a case the link should be deemed unavailable.

A. ACK-based TPC

One of the most straightforward TPC methods is based on counting acknowledgements or lack thereof for each transmitted packet. In ACK-based TPC (see Fig. 1) decision on changing the $\mathrm{P_{Tx}}$ is based solely on whether the ACK was received or not. Transmitter counts successful S and failed F transmissions. When the number of successes exceeds the predefined threshold $S_{\max}$ then $\mathrm{P_{Tx}}$ is decreased. When transmission fails and failure threshold is not exceeded then retransmission occurs. When F exceeds the threshold $F_{\max}$ and $\mathrm{P_{Tx}}<\mathrm{P_{max}}$, $\mathrm{P_{Tx}}$ is increased, F and S are set to zero, and transmission is retried. Finally, if transmitter failed to deliver the packet $F_{\max}$ times with $\mathrm{P_{max}}$, then the message is dropped.

B. RSSI-based TPC

RSSI provides information about average signal strength of the received packet. Since RSSI is measured by the receiver,

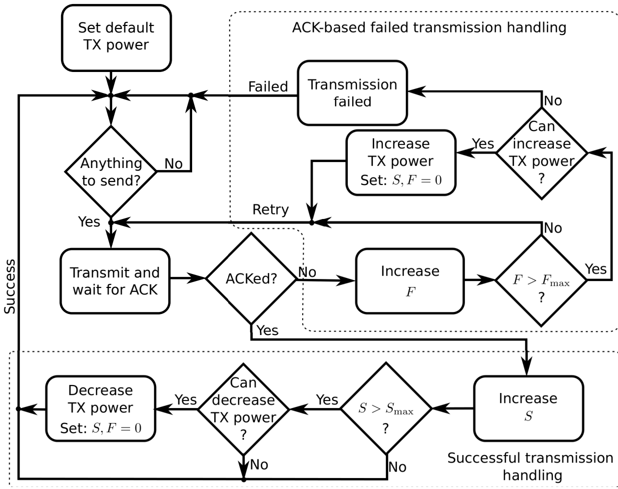


Fig. 1. Flowchart of ACK-based TPC algorithm

therefore the receiver has to either *i*) send the value with ACK or *ii*) send standard ACK while the transmitter measures the RSSI for the confirmation packet.

Since RSSI is only available for successfully transmitted packets, thus RSSI-based TPC algorithm is in large part identical to ACK-based TPC algorithm, but the RSSI values are compared against a desired range of [$\mathrm{RSSI}_{\min}$, $\mathrm{RSSI}_{\max}$] to ensure that for each transmission there is a power reserve that ensures successful transmission on one hand, and keeps P_{Tx} and power consumption low on the other.

III. PRR-BASED ALGORITHM

The aim of our PRR-based algorithm is to lower transmission power (radio signal power in transmitter) while still ensuring data packets are successfully delivered (i.e. PRR is on acceptable level).

Let's assume radio transmitter has k different $\mathrm{T_X}$ power settings P_i that imply different power consumptions e_i. Additionally, for each P_i we know the PRR (i.e. percentage of successfully transmitted packets). When node uses i-th P_{Tx}, then the expected overall power consumption E_i on a single packet depends on the number of transmission retries:

$$E_i = e_i p_i + 2e_i p_i q_i + \cdots + n e_i p_i q_i^{n-1} = e_i p_i \sum_{j=1}^{n} j q_i^{j-1} \quad (1)$$

where p_i is PRR for i-th P_{Tx}, $q_i = 1 - p_i$ and n is the maximal number of retries before packet is dropped. Term $p_i q_i^{j-1}$ in (1) represents the probability that $j-1$ transmissions failed and j-th transmission succeeded. Overall cost is thus equal to je_i, as a total of j transmissions were made. As n is large and $q_i \leq 1$, the sum in (1) is a geometric progression which yields:

$$\sum_{j=1}^{n} j q_i^{j-1} = \frac{\partial \left(\sum_{j=1}^{n} q_i^j \right)}{\partial q_i} \approx \frac{1}{(1-q_i)^2}. \quad (2)$$

Consequently, the energy (1) consumed on packet transmission using i-th P_{Tx} can be approximated as:

$$E_i \approx e_i p_i \frac{1}{(1-q_i)^2} = \frac{e_i}{1-q_i} = \frac{e_i}{p_i}. \quad (3)$$

TABLE I

PRR MEASURED FOR MICAZ NODES IN SEMI-URBAN AND OPEN FIELD ENVIRONMENTS OVER 20M TRANSMISSION WITH DIFFERENT $\mathrm{T_X}$ POWERS

$\mathrm{T_X}$ power setting	1	2	3	4	5	6	7	8
P_{Tx} – P_i [dBm]	-25	-15	-10	-7	-5	-3	-1	0
Energy cost – e_i [mW]	28.7	31.6	34.4	36.9	39.4	40.5	42.2	45.4
PRR semi-urban – p_i	0	0	0.95	1	1	1	1	1
e_i/p_i			**36.2**	36.9	39.4	40.5	42.2	45.4
PRR open field – p_i	0	0	0	0.22	0.75	0.89	0.93	0.95
e_i/p_i				167.7	52.5	45.5	**45.4**	47.8

Eq. (3) allows to assess power needed to deliver a single packet based on single transmission cost and current PRR value. P_{Tx} cannot be reduced extensively without affecting the PRR and increasing the costs.

Table I presents measured values of PRR for different $\mathrm{T_X}$ powers of MicaZ nodes over 20 m distance in semi-urban and open field environments. For semi-urban environment P_{Tx} set at P_3 minimises the cost of radio transmission. Even if 5 out of 100 transmissions fail ($p_i = 0.95$), and required retransmissions consume additional power, on average, savings on using lower P_{Tx} are not wasted. Similarly, for open field environment the best P_{Tx} is $\mathrm{P}_7 = -1\,\mathrm{dBm}$.

The PRR-based TPC algorithm starts with estimation of PRR for each P_{Tx} which is done by sending several data packets and monitoring the ACKs. Afterwards the transmitter uses estimated PRR to select optimal P_{Tx} – i.e. one that yields the smallest value of e_i/p_i. After a packet is sent (either acknowledged or not) the transceiver updates PRR estimation. If transmission failed then transceiver selects the optimal P_{Tx} and retransmits the packet. Packet is dropped if the number of retransmissions exceeds a predefined threshold.

One of the biggest challenges while implementing PRR-based TPC is to keep the PRR estimators updated. We have verified two approaches: *i*) PRR is updated after radio transmission only for the P_{Tx} used to transmit the packet, *ii*) PRR is updated periodically for all the $\mathrm{T_X}$ powers. In the first case PRR updates are based only on data transmissions, thus do not increase power consumption. The consequence is that until a $\mathrm{T_X}$ power setting is not used, the corresponding PRR is not updated. When propagation conditions worsen, the number of failed transmissions increases, thus lowering the PRR for current P_{Tx}. This increases estimated cost of the packet transmission using current power and finally leads to the situation when another $\mathrm{T_X}$ power setting is favored. When radio propagation improves, PRR for currently used P_{Tx} increases. Since PRR for other $\mathrm{T_X}$ powers is not updated, thus current P_{Tx} will remain the best power setting for minimising packet costs. Consequently, when channel improves, it is likely that P_{Tx} can be reduced, but the transceiver will ignore this and keep using current P_{Tx}.

The second approach eliminates this problem at the cost of additional communications. However, instead of calculating PRR for every P_{Tx} available we calculate PRR only for $\mathrm{T_X}$ powers of interest (i.e. those that are likely to be selected for future transmission). These powers are selected based on the following observations: *i*) PRR drops with P_{Tx} with exceptions

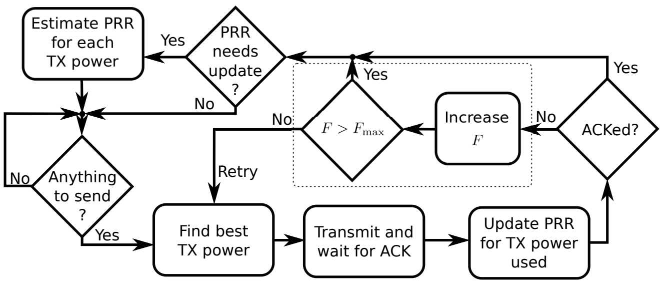
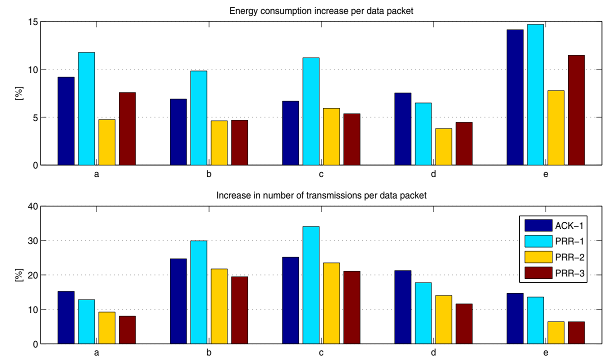


Fig. 2. Proposed PRR-based TPC algorithm



(a) semi-urban environment, fixed PRRs during test,
(b) semi-urban environment, variable PRRs, small number of random changes during tests (4 changes, each every 2 000 transmissions),
(c) semi-urban environment, variable PRRs, large number of random changes during tests (50 changes, each every 196 transmissions),
(d) open field environment, fixed PRRs during test,
(e) open field environment, fixed PRRs during test, step change of PRRs from 0 to 1 (no intermediate values).

Fig. 3. Results of numerical evaluation of ACK- and PRR-based TPC methods for different environments – energy increase in % relative to *oracle* algorithm

for PRR=1 and PRR=0 (cf. Table I), *ii*) PRR is calculated with fixed precision $\delta_0 = 0.01$, *iii*) for MicaZ nodes the smallest energy consumption on radio transmission $e_{\min} = 28.7\,\mathrm{mW}$ (cf. Table I), *iv*) for MicaZ nodes the median difference in power consumption between two successive $\mathrm{P_{Tx}}$ levels equals $\delta e = 2.5\,\mathrm{mW}$. The above observations allow to derive limits for PRR update that will define border values outside which determining the PRR is pointless as no value improves power consumption. Based on required energy to corresponding PRR rate, we can calculate the lower and upper bounds to be 0.11 and 0.92 respectively. First the transceiver updates PRR for $\mathrm{T_X}$ powers smaller than $\mathrm{P_{Tx}}$ currently used. This step ends when there are no more $\mathrm{T_X}$ powers or the PRR value drops to 0.1. In such case PRR evaluation is terminated and PRR value for smaller $\mathrm{T_X}$ powers is fixed to 0. In the second step PRR for higher $\mathrm{T_X}$ powers is updated until all $\mathrm{T_X}$ powers are updated or PRR value exceeds 0.92. When PRR exceeds 0.92 for some $\mathrm{P_{Tx}}$, then all higher $\mathrm{T_X}$ powers get a fixed PRR value of 1.

The difference compared to ACK-based TPC method is that here for every retransmission $\mathrm{P_{Tx}}$ level is used that is considered optimal.

IV. TPC EVALUATION

Evaluation of our PRR-based TPC algorithm was conducted using simulators implemented in MATLAB and OMNeT++ environments. Due to the simplicity of MATLAB simulator we have used it only to compare performance of ACK-based and PRR-based TPC algorithms. On the other hand, OMNeT++ was used to compare these two and RSSI-based method.

Four versions of TPC algorithms were implemented in MATLAB: simple ACK-based (denoted as ACK-1) and three versions of PRR-based (denoted as PRR-1,2,3). The ACK-based TPC algorithm is based on two thresholds $S_{\max}$ and $F_{\max}$ denoting number of successful and failed transmission that cause $\mathrm{P_{Tx}}$ to decrease and increase respectively. In MATLAB tests thresholds were set to $F_{\max} = 3$ and $S_{\max} = 20$ based on previous parameter space exploration. In PRR-1 PRR estimates are calculated and used as described in section III. In PRR-2 the PRR-3 estimates are updated every fixed number of transmissions, and the threshold $F_{\max}$ is set to large value, e.g. 50. PRR-3 adds hysteresis to $\mathrm{P_{Tx}}$: it is modified only if it reduces average energy cost by at least 0.1 mW.

We compare *average energy consumption* and *number of transmissions*. Since PRR-2 and PRR-3 algorithms use additional radio transmissions in order to estimate PRR values, therefore, comparison of algorithms is twofold. First, we take into account only transmissions required to transmit N data packets between two nodes. Second, we compare *average energy consumption* and *number of all transmissions* which includes also transmissions required to estimate PRR values. The above criteria are evaluated with respect to optimal values calculated as if the nodes knew exact PRR values (denoted as *oracle* algorithm), there was no randomness in the communication channel and every transmission was correctly received. Therefore, all values in Fig. 3 and Tab. III are given as increase (in percent) compared to optimal values. Results present mean and standard deviation over 100 tests, 10 000 packet transmissions each. PRR updates are periodically every 300 transmissions, according to the procedure presented in section III with at most 10 test transmissions per each $\mathrm{P_{Tx}}$.

In our simulations communication channel was modelled with normal probability distributions – mean values are equal to PRRs for each $\mathrm{P_{Tx}}$, while standard deviation was constant and equal to 0.15 (taken from OMNeT++ simulations). Exact values of mean for normal distribution were tested in real life experiments with MicaZ nodes in different environments (Tab. I, Fig. 3). The PRR update frequency and number of changes to the mean value of normal distribution was chosen so they were not synchronized.

It follows from Fig. 3 that PRR-based TPC allows to reduce the average energy cost of data packet transmission as well as reduce the number of radio transmissions. For example for semi-urban environment and fixed PRR the average energy consumption on PRR-2 is 4.4% smaller compared to ACK-1 algorithm. Also, for PRR-2 the average number of transmissions per data packet is smaller by over 6%, which results in decreased energy consumption and reduced interferences.

When additional radio transmissions required for PRR estimation are taken into account (methods PRR-2 and PRR-3), then average power consumption (on all transmissions) is

TABLE II
OMNeT++ RESULTS OF POINT-TO-POINT SIMULATION WITH ACK, RSSI AND PRR-BASED TPC METHODS

(a) ACK-based

S_{max}	TPM	CTM
3	1.67	15.26
9	1.30	12.57
15	1.18	11.91
21	1.13	11.68
27	1.10	11.68

(b) RSSI-based

RSSI [dBm]	TPM	CTM
-90 – -100	1.09	11.50
-87 – -97	1.07	11.31
-84 – -94	1.03	11.26
-81 – -91	1.01	11.58
-78 – -88	1.0013	12.53

(c) PRR-based

update interval	TPM	CTM
50	1.030	10.63
100	1.024	10.74
200	1.019	10.86
300	1.016	10.94

Legend: TPM – average Transmissions Per Message, CTM = average Cost of Transmitted Message

TABLE III
NUMERICAL EVALUATION OF ACK- AND PRR-BASED TPC ALGORITHMS ASSUMING PERFECT PROCEDURE FOR DETECTING CHANNEL CHANGES

Criteria			TPC method			
			ACK-1	PRR-1	PRR-2	PRR-3
Average energy consumption on data packets	SC	μ	6.93	9.79	5.16	**4.81**
		σ	1.70	3.98	3.45	1.25
	LC	μ	6.76	11.72	5.60	**4.94**
		σ	0.80	3.47	0.96	0.77
Average # of transmissions on data packets	SC	μ	24.13	29.14	21.56	**18.82**
		σ	5.70	14.95	11.90	7.53
	LC	μ	25.27	34.98	23.82	**20.96**
		σ	1.74	8.21	2.95	2.98
Average energy consumption on all transmissions	SC	μ	6.93	9.79	5.91	**5.54**
		σ	1.70	3.98	3.50	1.29
	LC	μ	**6.76**	11.72	11.82	11.11
		σ	0.80	3.47	1.18	0.93
Average # of transmissions on all transmissions	SC	μ	24.13	29.14	22.51	**19.74**
		σ	5.70	14.95	12.02	7.63
	LC	μ	**25.27**	34.98	31.59	28.64
		σ	1.74	8.21	3.20	3.08

SC/LC = small/large # of changes, σ = std. deviation, μ = mean value

smaller for ACK-1 algorithm. Still PRR-based TPC algorithms outperform ACK-1 in the number of transmissions but only when PRR settings are fixed (for communication channel with no changes of the PRR values over a time).

With effective PRR update procedure the PRR-based TPC algorithms have the ability to outperform ACK-based methods, even in changing communication conditions. To confirm this we have conducted additional test in which PRR updates were synchronised with changes in communication channel (as would be the case if perfect procedure for detecting channel changes was implemented). Results of this test show (Tab. III) that PRR-based TPC algorithms may outperform ACK-based both in terms of average energy costs and number of transmissions as long as communication channel doesn't change too often. When changes are frequent (as for the test with 50 random changes) then cost of periodic PRR update becomes comparable to data packet transmissions costs – for 50 changes PRR estimates are updated approximately every 170 packet transmissions and take up to 80 transmissions (8 T_X powers, 10 transmissions per each). For small number of changes (4) the total amount of transmissions required for PRR update accounts for less than 1 % of packets transmitted. For large number of changes (50) this rate soars to almost 8 %.

Further evaluation of TPC algorithms was conducted in OMNeT++ simulator with MiXiM framework. We developed new models, based on measurements of real devices [9], which allow to change transmission power and to draw variable amount of energy from battery depending on current T_X power. Results for classic ACK-based method are shown in Tab. II (a). These results concur with results obtained from MATLAB evaluation. This method performs best with small value of F_{max} parameter (around 3), and high value of S_{max} (around 20). The RSSI-based method gave slightly better results (Tab. II (b)). In this case if expected RSSI level grows, the probability of successful transmission rises, but above certain RSSI threshold the cost of a single transmission also goes up. The most promising is the algorithm based on PRR. In table II (c) we show simulation results of the PRR method. The cost of message transmission is lower than in the case of the other two algorithms.

V. Conclusions

It can be observed that there is no single TPC algorithm that outperforms all other. Depending on the channel and environment conditions, either PRR-2 or PRR-3 gives the best results. This suggests that it is difficult to use PRR-based algorithms that updates PRR estimates only based on data packets (as is the case for PRR-1 algorithm). Larger savings on packet transmissions can be achieved when PRR estimates are updated periodically. Unfortunately, periodic updates of PRR estimates affect energy consumption significantly. Therefore, although PRR-based TPC algorithms can save energy, they require comprehensive PRR update procedures that are not run periodically but are rather triggered when parameters of the radio channel change.

Acknowledgment

This paper has been accepted for publication in the proceedings of WSN-ADT Workshop that had been held in conjunction with NTMS 2014. It was supported by grant no. N 516 483740 from the Polish National Science Centre.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] T. Laukkarinen, J. Suhonen, T. Hamalainen, and M. Hannikainen, "Pilot studies of wireless sensor networks: Practical experiences," in *Design and Architectures for Signal and Image Processing (DASIP)*, 2011.

[3] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: adaptive transmission power control for wireless sensor networks," in *Proceedings of SenSys '06*. ACM, 2006, pp. 223–236.

[4] L. Correia and J. Nogueira, "Transmission power control techniques for MAC protocols in wireless sensor networks," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, Apr. 2008.

[5] J. Jeong, D. Culler, and J.-H. Oh, "Empirical analysis of transmission power control algorithms for wireless sensor networks," in *Networked Sensing Systems, 2007. INSS'07*, Jun. 2007, pp. 27–34.

[6] M. Kubisch, H. Karl, A. Wolisz, L. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *Wireless Communications and Networking*, vol. 1, Mar. 2003.

[7] L. H. Correia, D. F. Macedo, A. L. dos Santos, A. A. Loureiro, and J. M. S. Nogueira, "Transmission power control techniques for wireless sensor networks," *Computer Networks*, vol. 51, no. 17, 2007.

[8] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," in *IEEE SECON*, 2004, pp. 289–298.

[9] M. Słabicki, B. Wojciechowski, and T. Surmacz, "Realistic Model of Radio Communication in Wireless Sensor Networks," in *Computer Networks*. Springer, 2012, vol. 291, pp. 334–343.