Optimising WSN Battery Life using HARQ/TPC and Energy-Efficient Codes

Sunil Thunga, Shubhang Walavalkar, Vikas Kushwaha
Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India
8105319686 , 9731772881 , 9380798419
sunilthunga.221cs252@nitk.edu.in,
shubhangnwalavalkar.221cs348@nitk.edu.in,
vikaskushwaha.221cs260@nitk.edu.in

May 23, 2024

Abstract

A. Background of the problem statement

This paper aims on improving wireless sensor networks (WSNs) [1], which are becoming increasingly important in everyday applications such as healthcare, military and home automation [2]. The WSN's consists of sensor nodes equipped with a battery, a wireless communication unit, a processing unit, and a sensor unit. However, the widespread use of WSNs has drawbacks, especially in terms of energy limitations [3]. In wireless sensor networks (WSNs), nodes are typically powered by small batteries and are often deployed in harsh environments where batteries are difficult to replace or recharge [4]. This energy limitation greatly affects the efficiency and lifetime of WSNs. In order to manage energy constraints, this study emphasizes the significance of error reduction techniques that maximize energy consumption and prolong the lifetime of the network. The ultimate goal of this study is to demonstrate how we can further optimize the overall battery life of a WSN by utilizing the above mentioned techniques.

B. Challenges and issues of the problem statement

One major challenge faced by wireless sensor networks (WSNs) is their limited energy capacity, which arises from their dependency on sensor nodes that run on small batteries. Preserving optimal performance while extending network lifetime is a major concern. Because these networks are used for applications that require precise and highly accurate data collection, it can be challenging to retain accuracy when dealing with noise, changes in the environment, and faulty sensors [5]. Striking a balance between the energy economy and reliability is an important yet difficult trade-off. The energy consumption [6] of sensor nodes fluctuates during data transmission, reception, sensing, and processing, necessitating adaptive energy management. When wireless devices exist together, interference and congestion affect communication reliability [7]. Error correction algorithms need to be continuously improved in order to manage situations involving interference and congestion.

C. Existing approaches or methods and their issues

Continuous network operations in wireless sensor networks (WSNs) depend on effective energy management and overall energy utilisation. This uses modern components such as microcontrollers, communication modules, and low-power sensors. Together, these components reduce energy consumption for critical tasks such as capture, processing, and communication. However, the installation and purchase of such components are expensive and not feasible. Additionally, implementing communication protocols such as LoRaWAN and Zigbee [8], which are specifically designed for low-power, short-range applications, can limit network coverage in large-scale deployments but can improve overall energy optimization. However, striking a balance between communication frequency and the need for real-time data can be challenging. Currently, the implementation of sleep/wake cycles for sensor nodes can also increase network efficiency [9]. However this again may lead to increased latency in data transmission. This is an important way to conserve energy during times of low activity. Considering that WSNs are battery-powered networks, energy consumption has emerged as a critical factor in maintaining network

stability. Therefore, reducing energy is the key to extending the lifetime of WSNs [10]. Currently, there are two strategies to reduce battery consumption in WSN. Using forward error correction (FEC) control techniques is the first step. Unfortunately, FEC introduces overhead and drawbacks that cannot be tolerated in congested networks [11]. The costs associated with encoding and decoding complexity are handled by FEC. The second method is to use the Automatic Repeat Request (ARQ) error control module. In contrast to congested networks, this method works better on less congested networks. However, given the characteristics of WSNs, where networks are typically congested with hundreds of nodes and sometimes spread over large geographical areas, implementing the second technique seems inappropriate. Since each of these error controls (FEC and ARQ) has its own drawbacks, their implementation in WSNs does not seem to be completely applicable [12]. Furthermore, reducing node energy consumption relies heavily on the collaboration of energy-aware routing protocols such as hybrid energy-efficient [6] distributed clustering. Although activity is reduced, nodes can transition to a lower power state by adjusting their duty cycles based on dynamic network conditions. Furthermore, a careful balance is achieved by dynamically changing the quality of service parameters to effectively manage the trade-off between data accuracy and energy consumption.

D. Your problem statement

This project aims to improve battery life by integrating Hybrid Automatic Repeat request (HARQ), Transmit Power Control (TPC) [12], and advanced Energy-Efficient Codes [13], such as Low-Density Parity-Check (LDPC) or Reed-Solomon (RS) codes [1].

E. Objectives of the proposed work

The following are the objectives proposed by this paper:

1. Incorporating Hybrid Automatic Repeat Request (HARQ) and Transmission Power Control (TPC) in WSN's for battery optimisation. [12]

2. Integrating the above approach with error-correcting codes like Low-Density Parity Check (LDPC) and Reed Solomon (RS).

3. Researching which code (RS/LDPC) can better improve overall battery life in WSN. [1]

4. Increasing overall features of the WSN in addition to battery life leading to prolonged lifetime.

References

[1] A. Pal, The Internet of Things (IoT) – Threats and Countermeasures, https://www.cso.com.au/article/575407/internet-things-iot-threats-countermeasures/, [Accessed: 05-06-2020] (2018).

[2] S. Ragan, Here are the 61 passwords that powered the Mirai IoT botnet, https://www.csoonline.com/article/3126924/here-are-the-61-passwords-that-powered-the-mirai-iot-botnet.html, [Accessed: 05-06-2020] (2016).

[3] F. Paul, CWE-798: Use of Hard-coded Credentials, https://cwe.mitre.org/data/definitions/798.html, [Accessed: 05-06-2020] (2019).

[4] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, DDoS in the IoT: Mirai and Other Botnets, Computer 50 (7) (2017) 80–84. `doi:10.1109/MC.2017.201`.

[5] Mirai "internet of things" malware from Krebs DDoS attack goes open source, https://nakedsecurity.sophos.com/2016/10/05/mirai-internet-\of-things-malware-from-krebs-ddos-attack-goes-open-source/, [Accessed: 05-06-2020] (2016).

[6] E. Hayden, How hard-coded credentials threaten ICS security, https://searchsecurity.techtarget.com/tip/How-hard-coded-credentials-threaten-industrial-control-systems, [Accessed: 05-06-2020] (2018).

[7] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, Y. Zhou, Understanding the mirai botnet, in: 26th USENIX Security Symposium (USENIX Security 17), USENIX Association, Vancouver, BC, 2017, pp. 1093–1110.
URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[8] D. Strom, 9 ways to improve IoT device security, https://www.hpe.com/us/en/insights/articles/9-ways-to-make-iot-devices-more-secure-1701.html, [Accessed: 05-06-2020] (2017).

[9] S. Ciccone, Default Router Username and Password List, https://192-168-1-1ip.mobi/default-router-passwords-list/, [Accessed: 05-06-2020] (2019).

[10] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet scale iot exploitations, IEEE Communications Surveys Tutorials (2019) 1–1doi:10.1109/COMST.2019.2910750.

[11] I. P. Sam Edwards, Hajime: Analysis of a decentralized internet worm for IoT devices, https://security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf, [Accessed: 05-06-2020] (2016).

[12] Iot security– part 3 (101 – iot top ten vulnerabilities, https://payatu.com/iot-security-part-3-101-iot-top-ten-vulnerabilities/, [Accessed: 05-06-2020] (2018).

[13] S. Ciccone, Hardcoded Credentials: Why So Hard to Prevent?, https://www.veracode.com/blog/managing-appsec/hardcoded-credentials-why-so-hard-prevent, [Accessed: 05-06-2020] (2017).

****** END ******