Hybrid MQTT-COAP Protocol for Data Communication in Internet of Things

Emmanuel I. Nwankwo
Dept. of Telecommunications Engineering, School of Electrical Engineering and Technology, Federal University of Technology, Minna
Niger State, Nigeria
emmanueln_nike@hotmail.com

Elizabeth N. Onwuka
Dept. of Telecommunications Engineering, School of Electrical Engineering and Technology, Federal University of Technology, Minna
Niger State, Nigeria
onwukaliz@futminna.edu.ng

Michael David
Dept. of Telecommunications Engineering, School of Electrical Engineering and Technology, Federal University of Technology, Minna
Niger State, Nigeria
mikeforheaven@futminna.edu.ng

Zubair Suleiman
Dept. of Telecommunications Engineering, School of Electrical Engineering and Technology, Federal University of Technology, Minna
Niger State, Nigeria.
zubairman@futminna.edu.ng

***Abstract*—Wireless Sensor Networks (WSNs) consist mainly of resource constrained sensor nodes and gateways. Therefore, various lightweight communication protocols are emerging for Machine to Machine (M2M) communications. Among the various application layer protocols for data communication in WSNs, the two most popular protocols for constrained devices are the Message Queue Telemetry Transport Protocol (MQTT) with a variant for sensor nodes (MQTT-SN) and the Constrained Application Protocol (CoAP). Studies have shown that the performance of these different protocols are dependent on different network conditions. CoAP is more efficient in terms of message overhead and MQTT-SN is more efficient in terms of client complexity. Studies have further emphasized the levels of difficulties to implementing any of these protocols with regard to application requirements. In this paper we propose a hybrid MQTT-CoAP protocol technique using an abstraction layer that enables both MQTT-SN and CoAP protocol to be used in the sensor node. Performance evaluation of these protocols under the hybrid technique shows that the hybrid is feasible and while CoAP performs better in terms of energy consumption, the two protocols perform almost equally in latency. The observed values of latency and energy consumption in the developed hybrid technique was comparable to other studies.**

Keywords— *CoAP, Hybrid protocol, Machine to Machine Communication, MQTT-SN, Wireless Sensor Network*

I. INTRODUCTION

The Internet of things (IoT) being an emerging technology is comprised of a large number of different devices with diverse processing, sensing, actuating and communicating capabilities. Therefore, various new protocols are being developed in the protocol stack to facilitate communications in IoT environments [1]. In the IoT application layer, several protocols have been designed specifically for constrained devices and the two protocols of interest in this study include the Message Queue Telemetry Transport protocol for Sensor networks (MQTT-SN) and the Constrained application protocol (CoAP).

The Constrained application protocol (CoAP) was designed for use in devices with limited processing capability by the Constrained RESTful Environments (CoRE) working group of IETF [2]. It is a lightweight version of Hypertext transfer protocol (HTTP) that operates on the client server architecture using a subset of the HTTP methods making it interoperable with HTTP [3]. Unlike HTTP which runs over TCP, CoAP sends message compressed over UDP.

Message Queue Telemetry Transport protocol (MQTT) operates under the publish/subscribe architecture and runs over TCP just like HTTP. MQTT for sensor networks (MQTT-SN) [4] is a lightweight version of MQTT which is adapted to the peculiarities of the wireless communication environment for sensor nodes. MQTT-SN communicates using UDP and is designed in such a way that it is agnostic of the underlying network services. The major differences between the CoAP and MQTT-SN protocol is shown in Table 1.

TABLE I. MAJOR DIFFERENCES BETWEEN MQTT-SN AND COAP

	MQTT-SN	CoAP
Application Layer	Single Layer	Single layer with 2 conceptual sublayers (Message layer and Request Response Layer)
Reliability	3 Quality of Service (QoS) levels	Confirmable/Non-confirmable messages, Acknowledgments and retransmissions
Architecture	Publish/Subscribe	Request/Response, Resource/observe
Header Size	7 bytes	5 bytes

The MQTT Publish/Subscribe (pub/sub) messaging system [5] is an example of data-centric communication and is widely used in enterprise networks, mainly because of their support of a dynamic application topology and scalability. This nature of data-centric communication requires a broker to manage information between publishers and subscribers providing a higher level of abstraction for individual nodes, users or actors. Regardless of the numerous benefits and ease of this approach in WSN, it presents a single point-of-failure problem meaning that if the broker is down, communication in the whole network or the part of the network served by the broker is completely lost.

Message-centric communication on the other hand supported by CoAP greatly requires communication between individual units in the network and thus there is no single point of failure, however, it incurs some complexity in client node design. Moreover, Marti *et al* [6] noted that CoAP is preferable when security is stringent while MQTT-SN is preferred for speed. It was also noted in [7] that MQTT performs poorer than CoAP in terms of throughput and latency under high traffic and poor link conditions. The system engineer therefore has to decide between these two approaches the one that is best suited for the application requirements.

A hybrid MQTT-CoAP solution is proposed in this paper by developing an abstraction layer that enables the sensor node to support both the MQTT-SN protocol and the CoAP protocol concurrently. This solves the dilemma of the application developer as par which protocol to deploy for any constrained device. It means that, depending on the condition, a given sensor node can use any of the protocols for best results.

The rest of this paper is organized as follows: Section II discusses related works and their research findings, Section III provides a description of the developed abstraction layer and the API it provides as well as the algorithm employed. Section IV describes the experiment setup and the simulation environment as well as results from implementation. Section IV provides a discussion of the experimental results and then finally, Section V concludes the paper.

II. Related Works

No hybrid of CoAP and MQTT has been reported in literature, however, various authors have carried out comparative evaluation of both protocols. For example, authors in [8] evaluated the performance of MQTT and CoAP via a common middleware but it was implemented on devices with no resource constrains for running the protocols also classified as high-end IoT devices [9]. They found that when compared to CoAP, MQTT messages experience lower delays for lower link packet loss and higher delays for higher link packet loss.

The authors in [10] analyzed CoAP, MQTT, MQTT-SN and QUIC to understand the overheads of obtaining data from an IoT device at a sink to potentially disseminate it downstream. Their results show that CoAP is most efficient in terms of message overheads. Another study [6] on the performance evaluation of CoAP and MQTT-SN concluded that since MQTT-SN client nodes have less complexity compared to CoAP, MQTT-SN is more efficient than CoAP. They however stated that implementing MQTT-SN is rather inconvenient on an already deployed network and CoAP is simpler in terms of infrastructure.

Research has shown that there could be varying network conditions and other challenges such as bandwidth utilization, energy consumption, latency etc., in real life IoT scenarios and there is no single protocol that could be easily implemented and also be a panacea for all these challenges. Our hybrid MQTT-CoAP solution in resource constrained nodes would serve as a pragmatic step in deploying sensor nodes capable of running multiple application layer protocols simultaneously. This eliminates the tradeoffs that may be incurred when only a single protocol is implemented in the network.

III. Abstraction Layer

The current design and implementation of the abstraction layer which hybridizes the two protocols is such that it exposes APIs that abstracts the complexity of managing the two protocols individually as well as APIs that are suited to advanced users who intend to access core functionality of any single protocol. Fig 1 is a block diagram that illustrates the abstraction layer.

The abstraction layer provides APIs such as *Setup_Coap_Resource()*, *Init_Mqtt_Coap()*, *Mqtt_Sn_Pub()*, *Mqtt_Sn_Sub()*. The *Setup_Coap_Resource()* API is used to setup the CoAP resource method i.e. GET, POST, PUT or DELETE, endpoint URL and handler. The

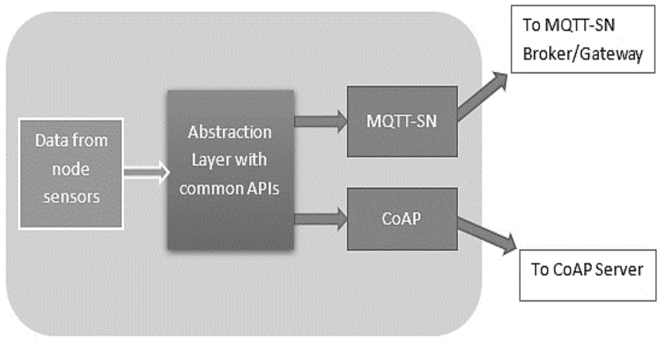

Fig. 1. Block diagram showing how the abstraction layer interfaces the protocols in a single device

Init_MQTT_Coap() API is used to initialize the CoAP resources that have been previously setup, create the MQTT connection to the broker and register any topics that would be used thereby converting them to numeric topics for MQTT-SN. Immediately after the MQTT-SN connects to the broker, it starts sending keep alive pings to the broker. The *Mqtt_Sn_Sub()* is specific to the MQTT-SN and it takes two arguments; the topic and the QoS. It is responsible for subscribing to a topic on the broker. The *Mqtt_Sn_Pub()* is also specific to the MQTT-SN protocol and it takes four arguments; the topic, the message, the QoS and a retain flag. It publishes a message on the broker on a specified topic which should have been previously registered during the initialization stage. The retain flag just tells the broker whether to retain the message for new subscribers or to discard it. The APIs in the abstraction layer has been used to determine the latency of MQTT-SN and CoAP protocols when they are both used simultaneously in the same constrained device.

IV. Experiment Setup

The performance of the system was evaluated when the abstraction layer was implemented on the constrained device. The performance of the protocols was measured in terms of latency per message size in bytes transmitted for different QoS levels in terms of MQTT and for CoAP. For MQTT, the latency was measured as the difference in time between when a message was published and when the subscriber received the message. For CoAP the latency was evaluated as the difference in time between when a request was sent to the node and when a response was received.

A. Simulation Environment Setup

The simulator used in this experiment is the Contiki OS based Cooja Network Simulator [11]–[13]. This is a firmware level simulator, which emulates the target device hardware thereby enabling execution of OS code and application compiled for the target platform. Using this simulator, timing-sensitive software can be tested [14]. Two types of sensor nodes were used, the Zolertia Z1 mote and the Skymote. To interface the motes and the RSMB (Really Small Message Broker), an IPV6 Router for Low-Power and Lossy Networks (RPL) Border router was setup on a Zolertia Z1 mote and using a tool called tunslip utility, a Serial Line Interface Protocol (SLIP) bridge was created between the RPL network and the local network. Fig 2 shows the interface for the simulation as well as introducing parts of the simulator.

The part labelled A is the network map. It shows the nodes added to the simulation as numbered circles in different colors. It also shows the IPV6 address of the node. 10 x 10 m^2 background grid for measuring distance, radio coverage area and other relevant information. The part labelled B shows the connection status of the border router to the local network via an open socket connection. The part labelled C is the simulation controls while D shows the mote debug/printed output in different color shades differentiating the nodes. The part labelled E shows the power tracker for the motes indicating the radio duty cycle for each mote.

Energy estimation was done using the Energest Module in the Cooja Network Simulator. Energest is a software-based mechanism that estimates energy by measuring the accumulated time the sensor node is in different states such as CPU, IRQ, LPM, Rx and Tx [15]. The average energy consumed in any state is calculated using equation (1), therefore the total energy consumed on average by the node in the intervals chosen is calculated using equation (2).

$$E_{avg}^{state} = (t_{end}^{state} - t_{start}^{state}) \times I_{state} \times V \quad (1)$$

$$\therefore E_{avg} = E_{avg}^{CPU} + E_{avg}^{Tx} + E_{avg}^{Rx} + E_{avg}^{LPM} \quad (2)$$

For the calculation to be done correctly, the current power consumptions of each state was obtained from the Sky mote datasheet [16].

B. Software Setup

Open source implementations of CoAP and MQTT-SN were CoAP for Contiki OS included in the example implementations and also the MQTT-SN for Contiki [17]. The implementations were modified in order to incorporate the development of the abstraction layer. This was compiled for both mote types, Zolertia Z1 and Skymote and placed according to the network topology in Figure 3. After organizing the nodes, the RSMB was started then the tunslip utility was run to bridge the RPL network and local network thereby enabling the nodes to communicate with the RSMB running on the local machine and listening for connections on port 1883.

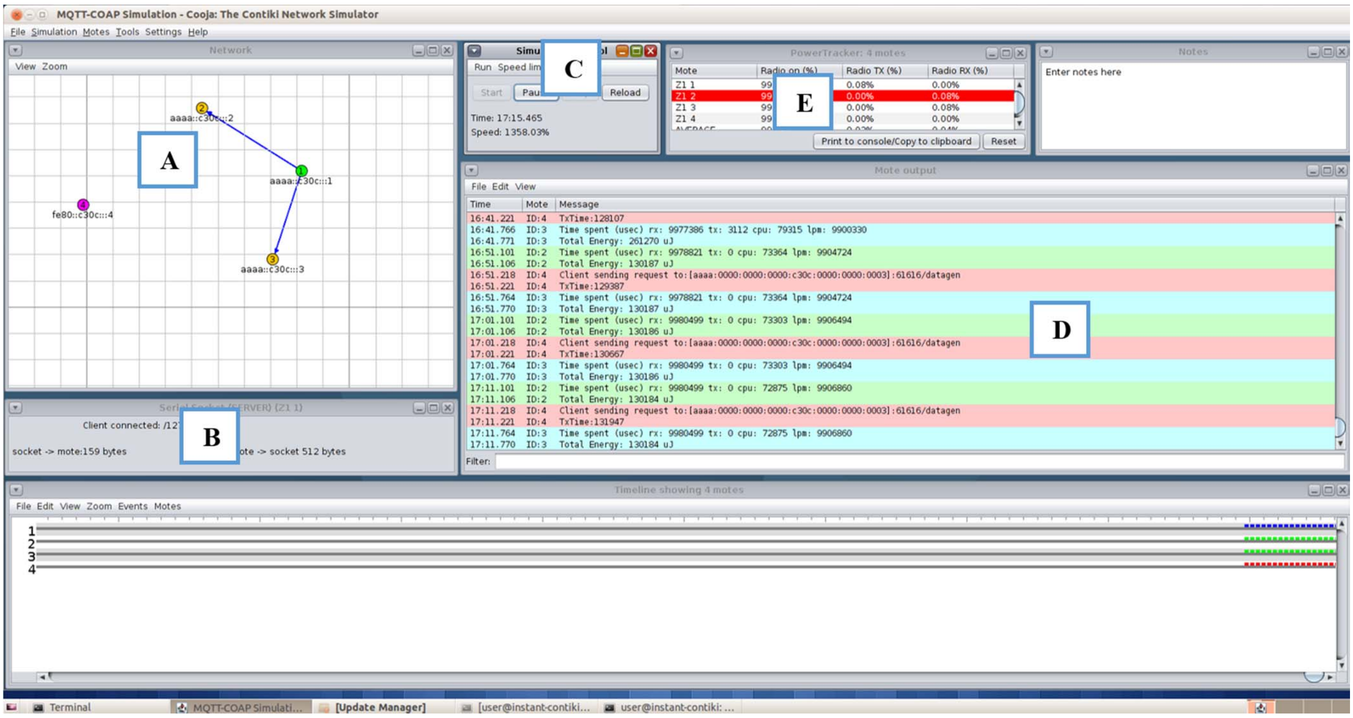

Fig. 2. Cooja Simulation environment showing the parts A-E as explained in section IV(A)

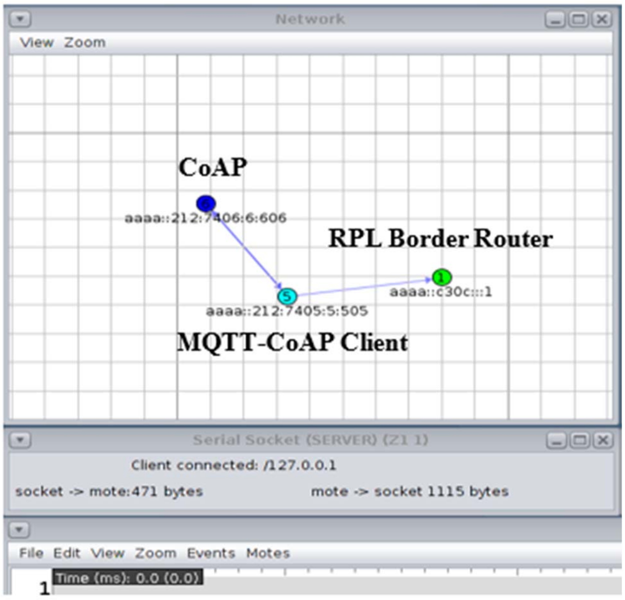

Fig. 3. Network topology for sensor connected nodes

The nodes were programmed to send packets in 10 s interval and to increase the packet size by 10 bytes after sending 10 packets starting from a packet size of 20 bytes.

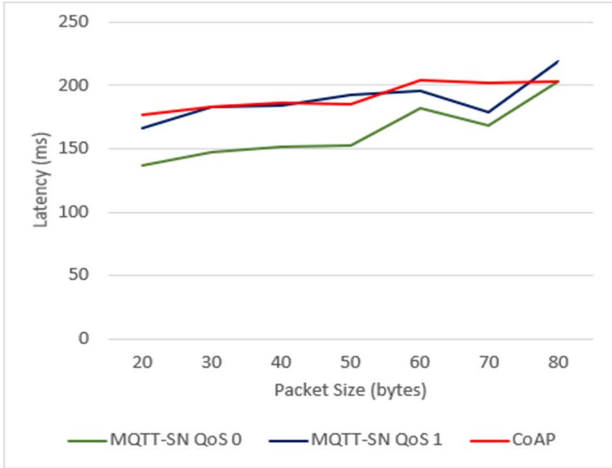

Fig. 4. CoAP and MQTT-SN Latency at different packet sizes (This shows that the abstractioon layer can manage the two protocols in a single constrained device without additional overhead.

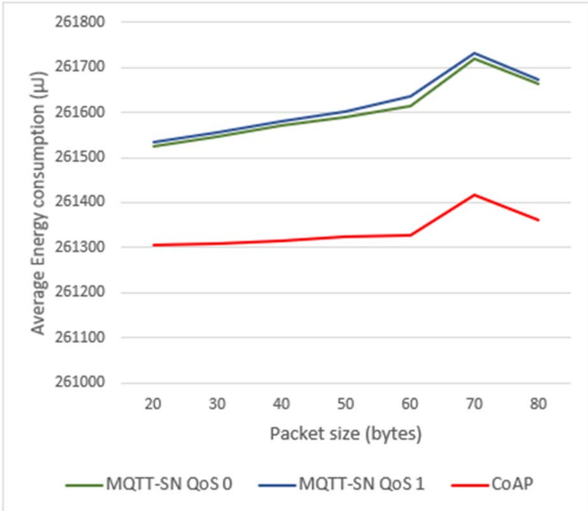

Fig. 5. Average energy consumption of CoAP and MQTT-SN transmitting and receiving different packet sizes (This, again, shows that the hybrid is feasible as the two protocols can function in a single constrained device via the abstraction layer.

V. Experimental Results

Since the experimental setup has only one publisher, one broker and one CoAP client, we experienced zero loss in message delivery. It was observed that the maximum packet size that could be transmitted from the node is 87bytes. This is due to the fact that the motes ran on Zigbee and this follows the IEEE 802.15.4 standard that has a limit on the data size allocated to user application [18].

The latency values were observed to slightly increase as the packet size increased. The lowest latency was observed in MQTT-SN QoS 0 while similar latency values were obtained for the CoAP and MQTT-SN QoS 1. The average latency was observed to be 163.2ms, 188.5ms and 191.5ms for MQTT-SN QoS 0, MQTT-SN QoS 1 and CoAP respectively. Fig 4 shows the graph of latency against packet size for MQTT-SN and CoAP.

Energy consumption of the node when using MQTT-SN for a single Tx/Rx operation in a 10s interval showed an average of 261.6mJ for both QoS 0 and QoS 1 while an average of 261.3mJ was observed for CoAP. The graph in Fig 5 shows the energy consumption for the different protocols when transmitting and receiving different packet sizes. Although it is not entirely clear as to why there are slightly irregular behaviors of latency and energy consumption at above 60 bytes, the MQTT-SN Specification [19] restricts maximum payload size to 60 bytes over ZigBee network due to the ZigBee network/APS layer.

VI. Conclusions and Future Work

In this paper, we proposed an abstraction layer that serves to create a hybrid of MQTT-SN and CoAP on constrained devices and evaluate the performance of the protocols when they are implemented together in a single constrained device using this technique. This will help to take advantage of both protocols on a single sensor node under varying application requirements. It is believed that this hybridization will add robustness to sensor nodes.

Our results reveal that when operating with MQTT-SN protocol in the hybrid system, average latency at an average data size of 50 bytes is 163.2ms for QoS 0, 188.5ms for QoS 1 while it is 191.5ms for CoAP. Also, the energy consumption for a single Tx/Rx operation in a 10s interval was an average of 261.6mJ, 261.3mJ for MQTT-SN and CoAP respectively. This work has shown that the hybrid of these two protocols is feasible and can greatly simplify application development for constrained devices.

As a significant improvement on this work, further studies can be done to leverage this technique and adaptively switch these protocols in situations of power saving, broker failures, varying link performances or to satisfy other desired network conditions. In a future work, multiple nodes with the hybrid will be introduced in a network to study its effects under such scenario. Also, poor network conditions will be simulated to study how the nodes will leverage the hybridized protocols.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the Nigerian Tertiary Education Trust Fund (TETFund) via the Institutional Based Research Intervention (IBRI) grant 2019.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] Z. Shelby, "Constrained Application Protocol (CoAP) draft-ietf-core-coap-17," 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-coap-17.

[3] A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, "Web services for the Internet of things through CoAP and EXI," *IEEE Int. Conf. Commun.*, pp. 1–6, 2011.

[4] A. Stanford-Clark and H. L. Truong, "MQTT for sensor networks (MQTT-SN) protocol specification," 2013.

[5] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003.

[6] Martí, Garcia-Rubio, and Campo, "Performance Evaluation of CoAP and MQTT_SN in an IoT Environment," *Proceedings*, vol. 31, no. 1, p. 49, 2019.

[7] F. Ouakasse and S. Rakrak, "A comparative study of MQTT and COAP application layer protocols via. performances evaluation," *J. Eng. Appl. Sci.*, vol. 13, no. 15, pp. 6053–6061, 2018.

[8] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *IEEE ISSNIP 2014 - 2014 IEEE 9th Int. Conf. Intell. Sensors, Sens. Networks Inf. Process. Conf. Proc.*, no. April, pp. 21–24, 2014.

[9] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, "A Review of Low-End, Middle-End, and High-End Iot Devices," *IEEE Access*, vol. 6, pp. 70528–70554, 2018.

[10] E. Liri, P. K. Singh, A. BIN Rabiah, K. Kar, K. Makhijani, and K. K. Ramakrishnan, "Robustness of IoT Application Protocols to Network Impairments," in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2018, pp. 97–103.

[11] M. Jevtić and N. Zogović, "Evaluation of wireless sensor network simulators," *Proc. 17th*, pp. 1303–1306, 2009.

[12] T. Mehmood, "COOJA Network Simulator: Exploring the Infinite Possible Ways to Compute the Performance Metrics of IOT Based Smart Devices to Understand the Working of IOT Based Compression & Routing Protocols," 2017.

[13] Y. Song, O. Zendra, and O. Zendra, "Using Cooja for WSN Simulations : Some New Uses and Limits To cite this version : Using Cooja for WSN Simulations : Some New Uses and Limits," 2016.

[14] J. Eriksson, "Detailed Simulation of Heterogeneous Wireless Sensor Networks," 2009.

[15] J. Schandy, L. Steinfeld, and F. Silveira, "Average power consumption breakdown of wireless sensor network nodes using IPv6 over LLNs," *Proc. - IEEE Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2015*, no. June, pp. 242–247, 2015.

[16] Moteiv Corporation, "Moteiv: tmote sky low power wireless sensor module," 2006.

[17] Â. I. da Silva, "MQTT-SN for Contiki," *Github*, 2016. [Online]. Available: https://github.com/aignacio/mqtt-sn-contiki_example. [Accessed: 29-Feb-2020].

[18] T. R. Burchfield, S. Venkatesan, and D. Weiner, "Maximizing Throughput in ZigBee Wireless Networks through Analysis, Simulations and Implementations," *Proc. Int. Work. Localized Algor. Protoc. WSNs*, pp. 15–29, 2007.

[19] A. Stanford-Clark and H. L. Truong, "MQTT for sensor networks (MQTT-SN) protocol specification," *Ibm*, p. 28, 2013.