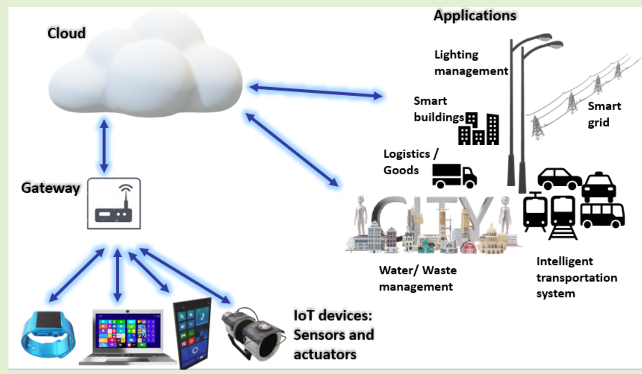
# Energy Efficient Data Compression in Cloud Based IoT

Halah Mohammed Al-Kadhim, *Member, IEEE*, and Hamed S. Al-Raweshidy, *Senior Member, IEEE*

***Abstract*—In this study, an adaptive data compression scheme (ADCS) is proposed for efficiently controlling the IoT device compression rate and energy consumption in the cloud based IoT network. The ADCS consists of two data compression schemes, the sensor Lempel–Ziv–Welch (S-LZW) scheme and the sequential lossless entropy compression (S-LEC) scheme. In Auto state, the ADCS can select the appropriate energy efficient data compression scheme for each IoT device, while taking into consideration the IoT device's processing capability, the available energy in each IoT device battery, and the amount of compression power. Our proposed scheme has been developed using mixed integer linear programming. The result verifies that the proposed ADCS scheme saves power by an average of 40% compared to the non-compression scheme (NCS) due to reducing the traffic load and the number of hops in the network, which leads to an ability to handle higher traffic demands and increasing the lifetime of IoT devices by 50% compared to NCS systems.**




***Index Terms*— Adaptive data compression scheme, energy efficiency, Internet of things, lossless compression, wireless sensor networks.**

## I. Introduction

IN RECENT years, cloud computing has attained huge popularity due to its vast storage and processing capabilities. However, IoT devices have limited energy and processing capabilities. Hence, IoT networks are integrated with the cloud environment to help in the storage and processing of data [1]. Energy efficiency is a critical aspect in IoT design and deployment, as IoT devices are usually battery-powered, and it is often difficult, expensive, or even dangerous to replace the batteries in many real physical environments such as recent and old buildings. Generally, more power is consumed in radio transmission and reception when compared to that of other node units (e.g. the microcontroller and memory) [2]. Effective data compression is imperative for reducing the power consumption of IoT devices.

Some data compression schemes are dedicated to IoT networks; in this paper, we used sensor Lempel–Ziv–Welch (S-LZW), [3], and sequential lossless entropy compression (S-LEC) [2]. These are lossless, energy efficient approaches with high compression ratios to minimize transmitted power. A smart building contains different electrical and electronic devices that can be monitored and controlled by smartphones or a PC, making modern buildings smart is a significant step towards a smart city that will enable future automation and optimization. Smart buildings also improve energy management by minimizing energy loss through the intelligent control of the high energy requirements of building devices, where smart building application incorporates IoT into their infrastructure.

Generally, IoT devices have limited hardware resources, such as energy, storage, and processing, which restrict the lifetime of a network. Hence, there is a need for efficient sensor network data compression schemes that do not consume high energy in processing or communication to decrease energy demands.

While we have already addressed some of the problems of the IoT (i.e. energy efficiency, reliability, and interference cancellation) in our previous work [1], [14], this work is complementary to our previous work [14]. It is aimed at achieving energy efficiency by minimizing the traffic power consumption of a cloud based IoT network. This energy efficiency was accomplished via two phases: radio power consumption and circuit power consumption. We formulated our model by using mixed-integer linear programming (MILP). MILP is a mathematical programming system — an optimization technique for getting an optimal solution to a function denoted as the

Manuscript received November 22, 2020; revised February 28, 2021; accepted March 2, 2021. Date of publication March 9, 2021; date of current version April 16, 2021. The associate editor coordinating the review of this article and approving it for publication was Dr. Arindam Basu. *(Corresponding author: Halah Mohammed Al-Kadhim.)*

The authors are with the Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, U.K. (e-mail: halah.alkadhim@gmail.com).

Digital Object Identifier 10.1109/JSEN.2021.3064611

objective function subject to a set of bounds and constraints. The MILP model has been developed to reduce the total traffic power of cloud based IoT network in two respects, as follows:

1. Radio power consumption minimization through:
- Optimizing the network path with a minimum number of hops and shorter link distance;
- Optimizing sub-channel selection by exploiting fading channel gain.

This part of the work was accomplished in our previous work [14]. The second part is addressed in the current paper:

2. Circuit power consumption minimization through:
- Optimizing the network power consumption through an adaptive data compression scheme (ADCS);
- The model adopts a suitable energy efficient data compression scheme for IoT devices while considering the device battery level, processing capability, and compression power;
- Optimizing the selection of IoT devices with minimum energy per bit and idle power.

The ADCS means we can use more than one data compression scheme and switch between them. In our work, we used two schemes: S-LZW and S-LEC. The energy consumption of ADCS is discussed and compared to the non-compression scheme (NCS) to show its effectiveness. Also, we proposed the ADCS in Auto state, which calculates how much power saving each data compression scheme would produce and selects the most energy efficient one (in our work, the S-LEC).

The contributions of this paper are summarised as follows:

- Design of a cloud based IoT network using a MILP model.
- Reduces network power consumption through ADCS using MILP optimization.
- Distributing traffic over the gateways to avoid traffic congestion to the cloud in the IoT network.
- Handling more traffic demands through data compression schemes.
- Maximizing the network lifetime.
- Minimizing the number of power ON devices.
- Optimize the selection of IoT devices that have the minimum energy per bit and idle power.

In the rest of this paper, Section II introduces the related work, in Section III, we describe the background to this research, and Section IV presents the proposed model of the adaptive data compression scheme. Section V introduces the objectives of the proposed model, and Section VI evaluates the model results. In Section VII, we state the conclusions of our research.

## II. Related Work

The notion of data compression has been around since the early days of computers [4]–[6] with many techniques for wireless sensor networks having been proposed recently to address the different restrictions and limitations of WSN [2], [7]–[10]. The goal of data compression is to minimize the amount of data to be transmitted over wireless channels. The format of the compressed data requires few bits, which leads to a minimization in the required inter-node communication, which considerably lessens the energy demand, thus extending the lifetime of an IoT device [11]. There are two components of a data compression task, as shown in Fig. 1. First, there is an encoding algorithm that converts a message into a compressed representation, which has the same data with as low length as possible. Secondly, a decoding algorithm reconverts the compressed representation into the original or nearly original message [12], [13]. Data compression techniques can be classified into two main types: lossy and lossless.

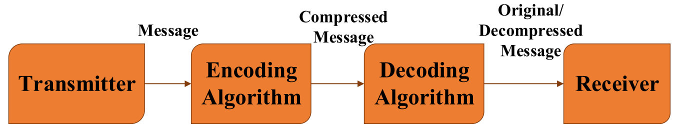



Fig. 1. Representation of data compression components.

Researches that achieving energy efficiency through adaptive data compression techniques include: The authors in [15] propose a system that uses surplus energy to compress data and/or expand the transmission range in an energy-harvesting WSN. An energy-adaptive data compression scheme is proposed in [16], to control the sensing rate in an energy-harvesting WSN. In the proposed scheme, by depending on the remaining energy of the node, each node can adjust the data collection period (to increase accuracy) and select the sensing rate without any rise in blackout time. In paper [3], nodes with surplus energy under a specific threshold compress data before transmission to reduce energy consumption, while nodes with surplus energy over the threshold transmit data without compression in order to decrease the delay time (latency) between nodes. In [17], a dynamic network selection mechanism that permits energy efficient and high quality patient health monitoring is introduced, it is targeting together a radio access network selection and adaptive data compression. In paper [18], An energy efficient data reduction scheme for IoT-Edge applications is proposed. A fast, error-bounded lossy compressor is proposed on the collected data prior to transmission. The transmitted data is rebuilt on an edge node and processed using supervised machine learning techniques.

Researches that achieving energy efficiency through developing data compression algorithms include: in paper [10], a tree-structured linear approximation scheme is proposed to compress sensing data according to an optimal rate-distortion (R-D) relationship. A novel data compression scheme is presented in [19], which enables a hybrid transmission mode for balanced data quality and power consumption. The proposed scheme encodes the raw data using a lossy technique, and the residual error from reconstruction is coded for lossless restoration. The goal of this study [20], was to develop a new coding scheme for delta compression as a technique for energy saving in an IoT environment. It is designed for applications where the sensor measurements are not needed in real time. Paper [21] presents a two-tier data reduction framework: the dual prediction scheme is used to reduce transmissions between cluster nodes and cluster heads, while the data compression scheme is used to reduce traffic between cluster heads and sink nodes. A fog-based optimized Kronecker-supported compression scheme is presented [22], that can achieve better compression results and reduce energy consumption in the industrial IoT (IIoT). A dynamic time division multiple access-based scheduling scheme is presented [23], which jointly considers energy consumption and data distortion. The trade-off between lifetime and distortion is examined, and the authors set up a framework that allocates the energy in every frame, determines the compression of the data to send along with the transmission durations, and performs power control. In this study [9], a data compression algorithm is presented with an error-bound guarantee for the WSN using compressing neural autoencoder networks. The proposed algorithm reduces data congestion and reduces energy consumption by exploiting spatio-temporal correlations in the training data to generate a low dimensional representation of the raw data. The adaptive rate-distortion feature balances the compressed data size (data rate) with the required error-bound guarantee (distortion level). A grade diffusion (GD) algorithm, along with the LZW compression technique, is used in [24], to increase the overall network lifetime, where the grade diffusion algorithm selects the most available energy node as the relay node in the routing process. Furthermore, the LZW algorithm minimizes the transmitting and receiving power by compressing the original data size. The authors in [8], explore the use of autoencoders as an efficient and computationally lightweight way to compress biometric signals. Although the presented techniques can be used with any signal showing a certain degree of periodicity, in this study they applied them to ECG traces, displaying quantitative results in terms of compression ratio, reconstruction error, and computational complexity.

In the literature, data compression is used to mainly focus on increasing accuracy, reducing latency, minimizing distortion level, spanning network lifetime, and modifying more robust data compression techniques. We developed our model with the purpose of achieving energy efficiency. However, it should be clarified that the model developed here differs from the existing ones in the following aspects: first, our energy efficiency model focuses on minimizing total network energy consumption contributed by various network components, such as sensors and routers. Second, in our energy efficiency model, in addition to the battery level that is used for optimally selecting the data compression scheme, the processing capability of the IoT devices is considered.

## III. Proposed Model of the Adaptive Data Compression System

In order to reduce data communication costs, there is a need to reduce the volume in the data representation. Data compression also reduces space requirements of compressed data of the storage hierarchy as well. Data compression can be implemented in either hardware or software. Consequently, data compression has the disadvantages of increasing the hardware and/or software complexity. Recent advances in lossless data compression algorithms include S-LZW and S-LEC, which are explained below.

LZW basics: LZW is a lossless, dictionary lookup-based algorithm that does not build its dictionary in advance, but rather, dynamically creates it based on the raw input stream [19].

TABLE I
A Compression String Table With Alphanumeric Character Strings That Are Encoded Into 12-Bit Codes

| Symbol String | Code |
|---|---|
| a | 1 |
| b | 2 |
| c | 3 |

TABLE II
LZW Compression Example

| Input symbols | ABABCBABABAAAAAAA | | | | |
|---|---|---|---|---|---|
| Current symbols | Next symbol | Output codes | | New string added to the table | |
| a | b | a | 1 | ab | 4 |
| b | a | b | 2 | ba | 5 |
| a | b | ab | 4 | | |
| ab | c | ab | 4 | abc | 6 |
| c | b | c | 3 | cb | 7 |
| b | a | ba | 5 | | |
| ba | b | ba | 5 | bab | 8 |
| b | a | ba | 5 | | |
| ba | b | bab | 8 | | |
| bab | a | bab | 8 | baba | 9 |
| a | a | a | 1 | aa | 10 |
| a | a | aa | 10 | | |
| aa | a | aa | 10 | aaa | 11 |
| a | a | aa | 10 | | |
| aa | a | aaa | 11 | | |
| aaa | a | aaa | 11 | aaaa | 12 |

It is a good fit for WSN because the dictionary structure permits it to generate various dictionaries according to the varied compressed contents and take advantage of repetition in the data. LZW replaces strings of characters with single codes in the dictionary. The algorithm sequentially reads in characters and finds the longest string $\omega$ that can be recognized by the dictionary. Then, it encodes $\omega$ using the corresponding codeword in the dictionary and adds the string $\omega$+k to it, where k is the character following string $\omega$. This process continues until all characters are encoded [20]. An example of this procedure is shown in table II. For simplicity, a three-character alphabet is used, as explained in table I.

Table I, is a compression string table with alphanumeric character strings that are encoded into a 12-bit code [25]. For this example, scarce letters, such as a, are assigned separately to the code. Repeated symbols that appear in long strings may surpass 30 characters in length. Hence, good compression is realized when a long string input is replaced by a 12-bit code, which achieves significant space saving. The LZW decompressor logically uses the same string table as the compressor and correspondingly constructs it as the input message code is translated.

*S-LZW algorithm:* S-LZW is a dictionary-based lossless compression algorithm used in resource-constrained WSN, because of its high compression ratio and lightweight. It is a modified version of the well-known LZW compression algorithm. It involves decreasing the weight of that algorithm, which has been used commonly in desktop PC environments [26], [27]. To adapt LZW into a sensor node, it needs to balance three main inter-related points: the size of the dictionary, the size of the data that need to be compressed, and the followed protocol when the dictionary fills. Most importantly, the memory restrictions require that LZW retains a dictionary size as small as possible. To decode a dictionary entry, however, all previous entries in the block must have received by the decoder. Unfortunately, the source never receives 100% of the sensor node data. To address this, the S-LZW algorithm separates the data stream into small, independent blocks, so only those following the lost packet are affected [3], [26].

*LEC algorithm:* LEC is a simple lossless entropy compression algorithm, which can be performed in a few lines of code and involves very low computational power while compressing data on the fly [28]. In addition, the used dictionary is very small, with its size being determined by the resolution of the analogue-to-digital converter. It is particularly suitable to be used on available commercial tiny sensors due to its low complexity and the small size memory needed for its execution. LEC is based on predictive coding [7], in which a predictor and an encoder are used. For a new data entry $x_i$ in a series, $\hat{x}_i$ is produced by a specified predictor, and the remainder $r_i = x_i - \hat{x}_i$ is calculated. This remainder $r_i$ is coded and then, sent to the receiving node. The differential predictor adopted in LEC is simple and popular, that is: $\hat{x}_i = x_i - 1$.

*S-LEC algorithm:* LEC has a general and non-adaptive predictor that cannot efficiently exploit temporal correlations for different WSN data streams for diverse WSN applications. To address this, a Sequential Lossless Entropy Compression (S-LEC), is proposed, which is a devised algorithm that extends the LEC to address its frailty of the shortage of robustness. S-LEC is capable of achieving highly robust compression performance for different sensor data streams; simultaneously, it enables energy-efficient employment and execution of resource-constrained WSN nodes in a relatively simple manner.

The performance of a compression scheme is usually evaluated by the compression ratio defined as the rate between the numbers of bits used to represent the transmitted after compression to the original data [2], [7], [9].

Table III shows the compression ratio and compression energy of both data compression schemes used in the model. S-LEC has a higher compression ratio and power consumption compared to S-LZW. Data compression increases the processing of the system due to the CPU cycles required for compression; consequently, data compression increases processing power consumption.

In this work, we have supposed smart buildings scenario with multiple user applications where the user application is performing in the cloud and requesting data collection. The data gathered by sensors in IoT devices, with IoT devices being connected to the cloud via the gateways.

Our proposed architecture of cloud based IoT is composed of three layers, as shown in Fig. 2.

TABLE III
A COMPARISON OF DATA COMPRESSION SCHEMES

| Data compression schemes | Compression Rate | Compression power mW/byte |
|---|---|---|
| S-LZW | 0.5101 | 0.00165 |
| S-LEC | 0.2793 | 2.897 |

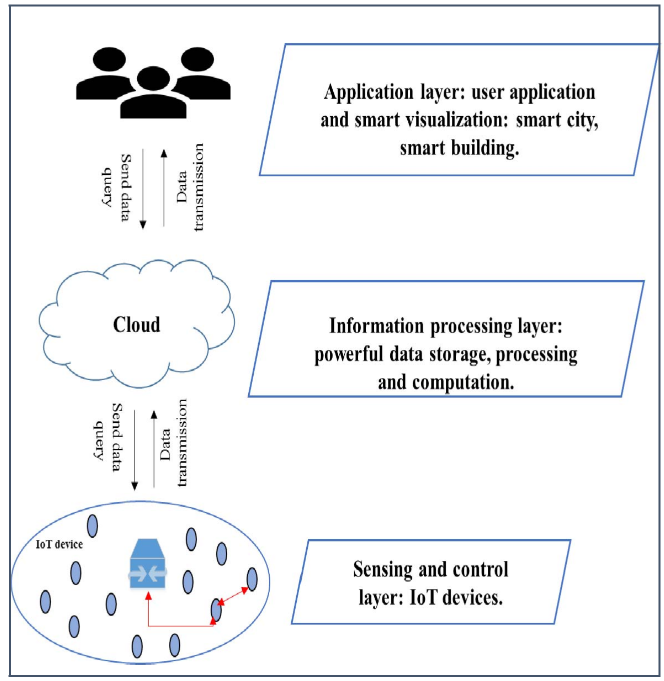



Fig. 2. Illustration of the proposed IoT physical network of a smart city.

1- Sensing and control layer, with the data being gathered in this layer and sent to the cloud for analysis.
2- Information processing layer, which comprises a data analytics center and storage media to process and analyze the unprocessed data.
3- Application layer, offering services to the end users by providing an interface for applications such as smart buildings [1], [14], [29].

We have supposed a cloud based IoT system where the IoT devices are split into one physical grid in smart buildings. Specifically, the grid comprises a number of IoT devices connected through a physical network separated across four buildings, as shown in Fig. 3.

Each smart building has a specific number of floors, and each building includes a particular number of IoT devices. The nodes on the first and second floor of each building serve as gateways to collect data to send to the cloud. Each IoT device is linked to its neighbors through a physical plan, with a star topology being proposed, as shown in Fig. 3.

Each IoT device has processing, storage, and functionality capabilities. We have supposed that each IoT device has two of the following functions: alarm, security, climate, entertainment. In the model, we have considered that each IoT device is connected to variant sensors with specifications, (i.e. functionality and location).

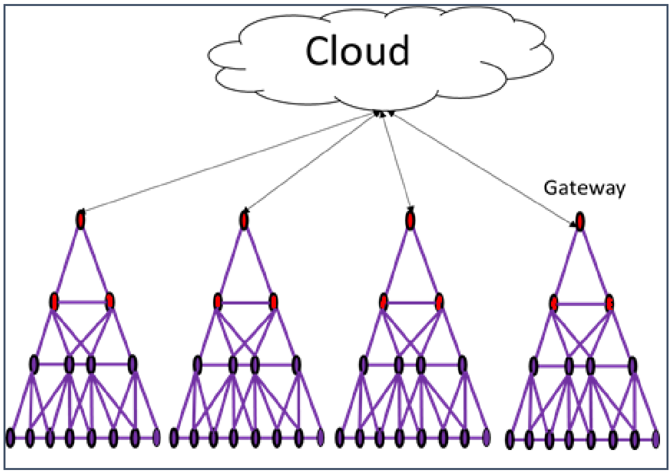



Fig. 3. Topology of one of the smart buildings in the proposed IoT network of a smart city.

In this cloud based IoT prototype, the IoT acts as the data source for the cloud, while users are the data requesters for the cloud. Cloud computing provides a platform as a service, which enables users to run, manage, and develop their applications. For example, during a data request from an application for real-time information, such as temperature or humidity, the application layer delivers the request to the cloud. Hence, the cloud will process the request and send the results to the application layer. For this purpose, the cloud will demand these data from the IoT devices located in the appropriate area and then gather information upon the gateways connected to it.

## IV. Objectives of the Proposed Model

The routing concept in this work is based on the flow conservation constraint for the traffic flows in the physical network by Tucker [30]. That is accomplished by creating a parameter $LK_G^d$ which indicates the traffic between the IoT device (d) and the cloud (G). Eq (1), as shown at the bottom of the page. It is also explained in our previous works [1], [14]. A binary variable $R_{ij}^{dG}$ is formed, which represents the route between the IoT device (d) and the cloud through the repeater's nodes (i, j) where j is neighbor of i.

$\forall$ d, i $\in$ D, d $\neq$ G

$$\{\sum_{j\in NB[i]} R_{ij}^{dG} - \sum_{j\in NB[i]} R_{ji}^{dG}\} = \mathrm{LK}_G^d \tag{2}$$

$$\{\sum_{j\in NB[i]} R_{ij}^{dG} - \sum_{j\in NB[i]} R_{ji}^{dG}\} = 0 \tag{3}$$

$$\{\sum_{j\in NB[i]} R_{ij}^{dG} - \sum_{j\in NB[i]} R_{ji}^{dG}\} = -\mathrm{LK}_G^d \tag{4}$$

It states that if the traffic flowing into a node is the same traffic flowing out of a node, then the node is not a source or a destination. If the traffic out of the node minus the traffic entering the node equals the demand originating in the node, then it is a source. If the traffic that enters it minus the traffic that leaves it equals the demand destined to it, then it is a destination as shown in Fig. 4.

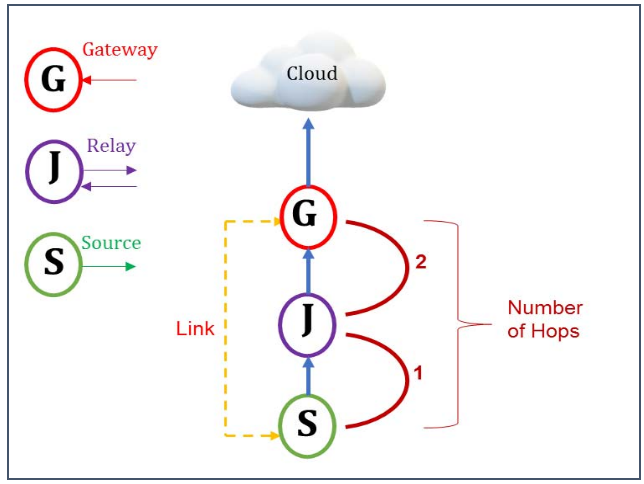



Fig. 4. Routing path between an IoT device and the cloud in the physical network.

TABLE IV
List of the Sets Used in the MILP Model

| Set | Description |
|---|---|
| D | Set of IoT devices. |
| A | Set of data compression schemes. |

TABLE V
List of the Variables Used in the MILP Model

| Variables | Descriptions |
|---|---|
| $CI_a^i$ | The IoT device with its corresponding compression scheme. |
| $NT_G^i$ | IoT device data traffic after compression. |
| $T_i$ | The power ON IoT devices. |

The model calculates the number of hops (H) in the network paths by the following equation:

$$\sum_{\substack{d\in D,\\ d\neq G}} \sum_{i\in D} \sum_{j\in D, i\neq j} R_{ij}^{dG} = H \tag{5}$$

where H: Variable

### A. Minimizing Power Consumption

Mathematically, our proposed model is as follows: the following constraint states the data traffic between the sensor node ($i$) and the cloud (G) after compression:

$$\sum_{a\in A} BT_G^i * CR_a * CI_a^i = NT_G^i \qquad \forall i \in D \tag{6}$$

where $BT_G^i$ is data traffic between an IoT device and the cloud before compression. $CR_a$ is the compression ratio; $CI_a^i$ is the IoT device with its corresponding compression scheme, and $NT_G^i$ is IoT device data traffic after compression.

We propose that the value of the IoT device bit rate should not exceed 10 Mbps. We have explained the key

$$LK_G^d = \begin{cases} 1, & \text{If there is link between the IoT device and the cloud} \\ 0, & \text{Else} \end{cases} \tag{1}$$

TABLE VI
LIST OF THE PARAMETERS USED IN THE MILP MODEL

| Parameters | Description |
|---|---|
| $\boldsymbol{CP_a}$ | The compression power for each compression algorithm in mW/byte |
| $\boldsymbol{E_i}$ | Energy per bit for each IoT device in mW/kbps. |
| $\boldsymbol{DL_i}$ | The idle power of each IoT device in mW. |
| $\boldsymbol{CR_a}$ | The compression ratio of the compression scheme. |
| $\boldsymbol{BT_G^i}$ | Data traffic between an IoT device and the cloud before compression. |
| $\boldsymbol{B_i}$ | The battery energy in joules of the IoT device. |
| $\boldsymbol{P_i}$ | The average power in joule required for transmission in the IoT device. |

notation used in the optimization model, which are listed in Tables IV and VI. The following restrictions evaluate the total traffic power in the network:

$$\text{Objective: Minimize} \quad \underbrace{\sum_{i\in D} CI_a^i * CP_a}_{\substack{\text{Processing/data}\\\text{compression power}}} + \underbrace{\sum_{i\in D} E_i * NT_G^i}_{\text{Transmission power}} + \underbrace{\sum_i T_{i\in D} * DL_i}_{\text{Idle power}} = TP \tag{7}$$

The values of the energy per byte $(E_i)$ and the idle power $(DL_i)$ are real ones taken from different IoT device data-sheets, namely, SPWF04SA, SPWF04SC [31], ESP32 [32], ESP8266EX [33], ZG2100M/ZG2101M Wi-Fi®Module Sheet [34], CC3100 SimpleLink™Wi-Fi®Network Processor, Internet-of-Things Solution for MCU Applications [35], CC3200MOD SimpleLink™Wi-Fi®and the Internet-of-Things Module Solution, a Single-Chip Wireless MCU [36].

### B. Maximizing Network Lifetime

It should be noted that providing energy efficiency in an IoT network does not guarantee that the battery lifetime of a device will be long. Hence, it is important to consider the energy level of the battery for each IoT device in the network before transmission and/or the compression process. The following constraint ensures that the residual energy in the IoT device battery will be at least 10% of the battery level:

$$B_i - E_i * NT_G^i - T_i * DL_i - \sum_i CI_a^i * CP_a \geq 0.1 * B_i \tag{8}$$

where $E_i$ is the energy per bit in joules and $DL_i$ is the idle power of each IoT device in joules, whilst the battery level is from 100 to 600 joules [37]–[39]. Thereby, this constraint

TABLE VII
EVALUATION PARAMETERS

| Parameter | Parameter value |
|---|---|
| Number of buildings | 4 |
| Number of sensor nodes per building | 15 |
| Number of gateways per building | 3 |
| Number of floors per building | 4 |
| Capacity limit | 10 Mbps |
| Radio communication standard | IEEE 802.11 |

TABLE VIII
POWER CONSUMPTION OF DIFFERENT IoT DEVICES

| IoT device | EI (mW/kbps) |
|---|---|
| CC3100 | 0.08 |
| ESP8266EX | 0.051 |
| ZG2100M/ZG2101M | 0.0462 |
| ESP32 | 0.0615 |
| SPWF04SA/SPWF04SC | 0.108 |
| CC3200 | 0.0762 |

diminishes the threat of node failure and, hence, extends the network lifetime. The following constraint evaluates the number of sub-operations of each IoT device:

$$\frac{B_i}{P_i} = N_i \tag{9}$$

The sub-operation is the number of the tasks of the node during its lifetime before it drops out.

## V. RESULT AND DISCUSSION

In order to evaluate the performance of our work, we considered a smart buildings scheme (i.e. hospital or shopping center) where the physical layer of 60 IoT nodes connected by 136 bidirectional wireless links. These IoT nodes were distributed randomly in buildings.

In this section, we present the numeric results of our MILP model. We have considered different types of IoT devices with variant levels in terms of power consumption, where detailed evaluation parameters are summarized in Table VII. The power consumption of used IoT devices is displayed in Table VIII.

We aim to answer the following questions: (1) How does the ADCS system affect the network in terms of energy efficiency? (2) What is the influence of the ADCS system on the IoT network as the data traffic increases? (3) What is the number of hops in the network in the ADCS and NCS systems? (4) How long is a node's lifetime in the network? To answer these questions, we ran the model for different scenarios of optimization, as follows: The first involved the non-compression NCS, whilst the ADCS includes the S-LEC and S-LZW compression schemes and the (Auto) scheme, in which the ADCS optimally selects the most energy efficient data compression scheme according to (3).

The results in Fig. 5 display the total power consumption of the IoT network in mW, versus the different percentages of the number of IoT devices that generate 500 kbps of the bit rate for each device, where an individual scheme has been selected manually each time. The results display that there is an average power saving of 33% in the S-LZW compression

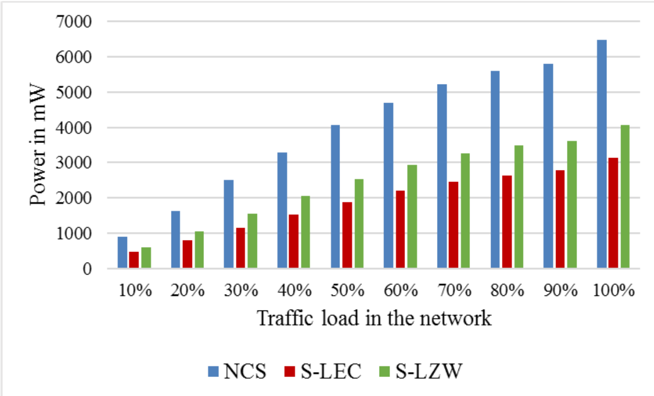



Fig. 5. Total power consumption in the IoT network when the link bit rate is 500 kbps for each node.

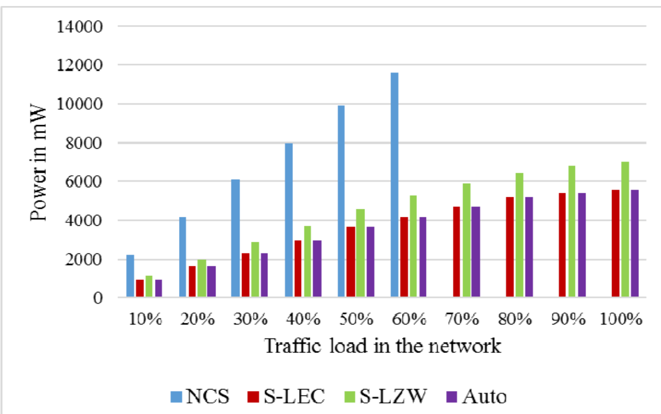



Fig. 6. Total power consumption in the IoT network when the link bit rate is 1000 kbps for each node.

scheme and approximately 40% in the S-LEC compression scheme compared to NCS, due to reducing the traffic of each link in addition to selecting efficient energy per bit and idle power IoT devices, as in (7).

Fig. 6 displays the total power consumption of the network in mW, with a 1000-kbps bit rate for each device. They show that the network is fully working in the NCS as long as the traffic load is below 60%. However, when it rises above this level, the network goes down with NCS due to packet dropout and, hence, data compression should be used to minimize the traffic. Additionally, the Auto scheme automatically (adaptively) chooses the S-LEC scheme as the optimal selection since it minimizes the transmission power more than S-LZW does.

Fig. 7 displays the number of hops in the network with a bit rate of 1000 kbps for each node. The number of hops has been estimated from the flow conservation constraint that is explained extensively above and in our former works in [1], [14]. The number of hops in the NCS is higher, being almost 60 hops at 60% of traffic load. While in data compression, the network does not need multiple routes, and so it can use the same number of hops without the network breaks down.

Fig. 8. displays the number of sub-operations during the node lifetime with different traffic load in the network. The

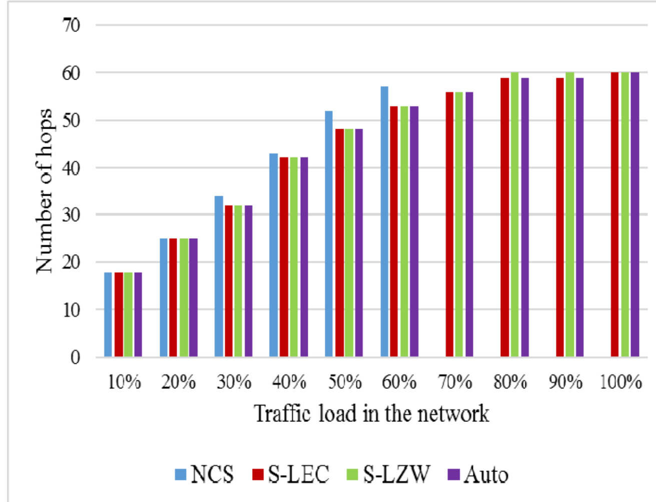



Fig. 7. Total number of hops when the link bit rate is 1000 kbps for each node.

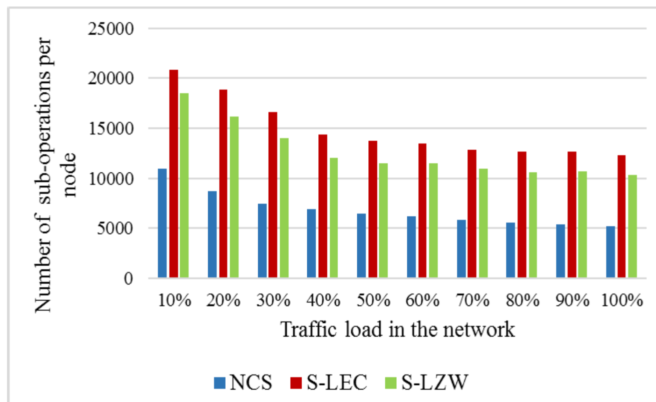



Fig. 8. Number of sub-operations for each node when the link bit rate is 500 kbps for each node.

results show that the number of sub-operations decreases as the traffic load increases. Furthermore, in ADCS, the number of sub-operations increased to 50% compared to NCS.

Future data compression schemes can be added to the system which may have better performance than S-LEC, the important issue here is that ADCS optimizes equation (7) through MILP for automatically (adaptively) selecting the optimal energy efficient scheme.

## VI. Conclusion

We have developed a MILP model using S-LEC and S-LZW data compression schemes to reduce the total traffic power consumption in the cloud based IoT network. The proposed scheme ADCS can select the most energy efficient data compression scheme, while also considering the IoT device battery level, processing capability, and compression power. The S-LEC and S-LZW data compression schemes are compared with a non-compression scheme, and the results showed an energy saving of 33% and 40% for S-LZW and SLEC, respectively, compared to NCS. That is a result of reducing the number of hops and the traffic load in the network; consequently, the system is able to handle a higher traffic demand, and this ability expands the IoT device lifetime by 50% compared to the NCS system.

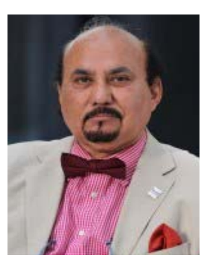
## Acknowledgment

The author Halah Mohammed Al-Kadhim would like to thank the Higher Committee for Education Development in Iraq (HCED) for funding her Ph.D. scholarship.

## References

[1] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient and reliable transport of data in cloud-based IoT," *IEEE Access*, vol. 7, pp. 64641–64650, 2019.
[2] Y. Liang and Y. Li, "An efficient and robust data compression algorithm in wireless sensor networks," *IEEE Commun. Lett.*, vol. 18, no. 3, pp. 439–442, Mar. 2014.
[3] M. J. Kang, S. Jeong, I. Yoon, and D. K. Noh, "Energy-aware determination of compression for low latency in solar-powered wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 2, Feb. 2017, Art. no. 155014771769416.
[4] J. S. Vitter, "Design and analysis of dynamic Huffman codes," *J. ACM*, vol. 34, no. 4, pp. 825–845, Oct. 1987.
[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 337–343, May 1977.
[6] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
[7] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 411–413, Jun. 2008.
[8] D. Del Testa and M. Rossi, "Lightweight lossy compression of biometric patterns via denoising autoencoders," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2304–2308, Dec. 2015.
[9] M. Abu Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Rate-distortion balanced data compression for wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 12, pp. 5072–5083, Jun. 2016.
[10] C.-M. Wang, C.-C. Yen, W.-Y. Yang, and J.-S. Wang, "Tree-structured linear approximation for data compression over WSNs," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2016, pp. 43–51.
[11] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 37–59, Jan. 2012.
[12] G. E. Blelloch, "Introduction to data compression," Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 2001.
[13] A. A. Preethi and A. Ramakrishnan, "A survey on data compression techniques in wireless sensor network," *SSRG Int. J. Electron. Commun. Eng. (SSRG-IJECE)*, vol. 2, no. 3, Mar. 2015.
[14] H. Al-Kadhim and H. Al-Raweshidy, "Energy efficient traffic in cloud based IoT," in *Proc. Int. Conf. Ind. Internet Things Smart Manuf. Imperial College London*, London, U.K., 2018.
[15] J. M. Yi, E. Ji Oh, D. K. Noh, and I. Yoon, "Energy-adaptive data compression and transmission range determination for energy-harvesting wireless sensor networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2017, pp. 383–385.
[16] I. Yoon, J. M. Yi, S. Jeong, J. Jeon, and D. K. Noh, "Dynamic sensing-rate control scheme using a selective data-compression for energy-harvesting wireless sensor networks," *IEMEK J. Embedded Syst. Appl.*, vol. 11, no. 2, pp. 79–86, Apr. 2016.
[17] A. A. Abdellatif, A. Mohamed, and C.-F. Chiasserini, "User-centric networks selection with adaptive data compression for smart health," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3618–3628, Dec. 2018.
[18] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient IoT data compression approach for edge machine learning," *Future Gener. Comput. Syst.*, vol. 96, pp. 168–175, Jul. 2019.
[19] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for IoT," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 245–254, Apr. 2017.
[20] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient IoT solutions," in *Proc. 25th Telecommun. Forum (TELFOR)*, Nov. 2017, pp. 1–4.
[21] A. Jarwan, A. Sabbah, and M. Ibnkahla, "Data transmission reduction schemes in WSNs for efficient IoT systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1307–1324, Jun. 2019.
[22] S. Chen, Z. Wang, H. Zhang, G. Yang, and K. Wang, "Fog-based optimized kronecker-supported compression design for industrial IoT," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 1, pp. 95–106, Jan. 2020.
[23] A. Biason, C. Pielli, A. Zanella, and M. Zorzi, "Access control for IoT nodes with energy and fidelity constraints," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3242–3257, May 2018.
[24] S. Prabu and R. Maheswari, "Improving wireless sensor network lifespan by efficient routing algorithm and LZW compression," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–5.
[25] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, Jun. 1984.
[26] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. -SenSys*, 2006, pp. 265–278.
[27] Z. Yan-li, F. Xiao-ping, L. Shao-qiang, and X. Zhe-yuan, "Improved LZW algorithm of lossless data compression for WSN," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Jul. 2010, pp. 523–527.
[28] C. P. Antonopoulos and N. S. Voros, "A data compression hardware accelerator enabling long-term biosignal monitoring based on ultra-low power IoT platforms," *Electronics*, vol. 6, no. 3, p. 54, Jul. 2017.
[29] N. Kaur and S. K. Sood, "An energy-efficient architecture for the Internet of Things (IoT)," *IEEE Syst. J.*, vol. 11, no. 2, pp. 796–805, Jun. 2017.
[30] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 176–186, Jun. 2009.
[31] STMicroelectronics, *SPWF04SA, SPWF04SC Standalone and Serial-to-Wi-Fi B/G/N Intelligent Modules*, document DocID029757 Rev 5, Nov. 2017.
[32] Espressif Systems, Shanghai, China. (Jun. 2017). *ESP32 Datasheet*. [Online]. Available: https://www.rototron.info/wp-content/uploads/PiESP32_Datasheet.pdf
[33] Espressif Systems, Shanghai, China. (2018). *ESP8266EX Datasheet, Version 5.8*. [Online]. Available: https://datasheetspdf.com/datasheet/ESP8266EX.html
[34] Microchip Technology, Chandler, AZ, USA. (2010). *G2100M/ZG2101M Wi-Fi Module*. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/70624a.pdf
[35] *CC3100 SimpleLink Wi-Fi Network Processor, Internet-of-Things Solution for MCU Applications*, document SWAS031D, Texas Instruments, Feb. 2015.
[36] *CC3200MOD SimpleLink Wi-Fi and Internet-of-Things Module Solution, a Single-Chip Wireless MCU*, document SWRS166, Texas Instruments, Dec. 2014.
[37] S. Mao, M. H. Cheung, and V. W. S. Wong, "An optimal energy allocation algorithm for energy harvesting wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 265–270.
[38] D. Hasenfratz, A. Meier, C. Moser, J.-J. Chen, and L. Thiele, "Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput.*, 2010, pp. 19–26.
[39] S. Peng and C. Ping Low, "Energy neutral routing for energy harvesting wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 2063–2067.

**Halah Mohammed Al-Kadhim** (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics and communication engineering from Baghdad University, Baghdad, Iraq, in 2008 and 2012, respectively, and the Ph.D. degree in electronic and electrical engineering from Brunel University London, U.K. Her research interests include the IoT, cloud computing, energy efficiency, reliability, adaptive data compression, interference cancellation, fading channel gain, traffic power consumption, and smart city applications.



**Hamed S. Al-Raweshidy** (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the Post Graduate Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991. He was with the Space and Astronomy Research Centre, Baghdad, PerkinElmer, Waltham, MA, USA, British Telecom, London, U.K., Oxford University, Oxford, U.K., Manchester Metropolitan University, Manchester, U.K., and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network Communications Centre, Brunel University London.