Exploring Tradeoffs between Energy Consumption and Network Performance in Cellular-IoT: a Survey

Nicholas Accurso*, Nicholas Mastronarde†, Filippo Malandra‡
Department of Electrical Engineering, University at Buffalo, Buffalo, NY
Email: *naccurso@buffalo.edu, †nmastron@buffalo.edu, ‡filippom@buffalo.edu

***Abstract*—Recent growth in the Internet of Things (IoT) has been remarkable. Among the solutions to accommodate such a growth is Cellular IoT (C-IoT), comprising a group of technologies extended from legacy cellular infrastructures. One of the key goals of C-IoT technologies is to extend the battery life of UEs (User Equipment) in the network. However, this often comes at the cost of degrading network performance. This work attempts to identify, categorize, and analyze the available literature on this problem. The literature is broadly categorized into three sections: scheduling, data processing, and sleep modes. In each of these sections, the literature is further sub categorized. Finally, a direction for future research is identified and discussed.**

I. Introduction

The Internet of Things (IoT) has seen tremendous growth over recent years. Gartner estimates that there were approximately 5.81 billion endpoints in the IoT in 2020 (21% increase from 2019) [1]. The applications that contribute to this growth are incredibly diverse in nature. From utilities monitoring to healthcare to agriculture, many such sectors account for a massive number of IoT endpoints requiring a reliable and efficient infrastructure to support their communication.

A variety of Low Power Wide Area Network (LPWAN) technologies (e.g., LoRaWAN and SigFox) have been developed solely to accomplish such tasks. These protocols can provide many benefits to an IoT network. Energy consumption due to communication is reduced drastically, and they are capable of accommodating ranges of over 10-15 km. However, in order to achieve such a large coverage area, they sacrifice data rate. For example, SigFox only achieves a data rate on the order of a few hundred bytes per second.

To find additional solutions to accommodate IoT growth, we look to existing cellular infrastructures, such as LTE. These technologies have already proven they are capable of dealing with a large amount of traffic over a wide coverage area with a much greater data rate than LoRaWAN or SigFox. But IoT devices cannot simply be placed in existing cellular infrastructures. In LTE, an entire Resource Block (RB) (180 kHz bandwidth) is allocated for each transmission, so many resources would go to waste for IoT communication, and network performance would deteriorate significantly as a result.

Newer protocols, namely LTE-M and NarrowBand IoT (NB-IoT), stem from these existing cellular infrastructures and create the foundation of Cellular Internet of Things (C-IoT). For instance, while LTE allocates an entire RB per transmission, NB-IoT allows multiple users to share resources within a single RB. This in turn allows more User Equipments (UEs) to exist in the same cell and share the same resources. While NB-IoT does increase delay, the reduction in required bandwidth, reduction in energy consumption, and increase in Maximum Coupling Loss (MCL) make NB-IoT a clear choice for delay-insensitive IoT applications. The general C-IoT architecture is illustrated in figure 1.

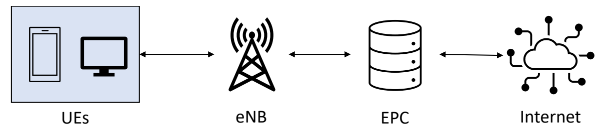

Fig. 1. "Cellular IoT Network Architecture" inspired from [2]

C-IoT networks tend to serve geographically distributed battery-operated sensors with a fixed energy budget. Because of this, a great deal of research has gone into improving the energy efficiency of these devices and in turn increasing their battery lifetimes. Doing so has three direct consequences: The amount of waste generated as a byproduct of the device's operation is reduced, maintenance of devices is decreased, and the cost of operation is reduced. By increasing the lifetime of batteries in the network, fewer batteries must be purchased over time, thus decreasing the cost of maintaining the network.

However, increasing energy efficiency often results in degraded performance. An example of this tradeoff is evident when comparing LTE-M with NB-IoT. While NB-IoT outperforms LTE-M on energy efficiency, it sacrifices important network performance metrics, such as delay and throughput. LTE-M on the other hand performs better in terms of delay and throughput, but is less energy efficient than NB-IoT.

In this survey, we identify, categorize, and analyze various methods for saving energy in C-IoT networks. While there are many surveys on C-IoT, there are none that focus solely on energy and the energy-performance tradeoff to the best of our knowledge. The rest of this survey is organized as follows: Section II provides an overview of relevant C-IoT protocols. Section III identifies similar existing surveys. Sections IV, V, and VI discuss energy saving technologies in the areas of scheduling, data processing, and sleep modes, respectively. In section VII, the conclusions of the survey are discussed.

II. Preliminaries

This section introduces NB-IoT and LTE-M, two novel C-IoT protocols designed to accommodate typical IoT traffic.

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685313

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

A. NB-IoT

NB-IoT was designed to provide an efficient way to integrate certain IoT devices with existing LTE networks. While LTE was the starting point for NB-IoT, it deviates from LTE in a number of ways to provide a more efficient use of resources, lower power consumption, and extended coverage. In LTE, UEs are assigned whole RBs (180 kHz) for each transmission. For UEs not utilizing the entirety of this RB, resources are wasted. NB-IoT, by contrast, can assign multiple transmissions in each RB, drastically reducing wasted resources for the typical low data transmission of many IoT applications. These resources can be assigned in 3 different modes: In an existing LTE band, in the guard band between LTE bands, and standalone in the GSM spectrum.

Uplink in NB-IoT can be either single-tone or multi-tone. Multi-tone transmissions use Single Carrier Frequency Division Multiplexing (SC-FDMA) with 15 kHz subcarrier spacing. Single-tone transmissions can have either 15 kHz subcarrier spacing with 0.5 ms slots, or 3.75 kHz subcarrier spacing with 2 ms slots. Downlink in NB-IoT uses Orthogonal Frequency Division Multiple Access (OFDMA) with 15 kHz subcarrier spacing and 1 ms subframes.

NB-IoT supports several features that allow UEs to save energy. Discontinuous Reception (DRX) allows a UE to turn off its Radio Frequency (RF) circuitry for short periods of time (up to a few seconds long). Once this time has expired, the device's RF circuitry is turned back on, and the device monitors the radio control channel for paging information. The periods of sleeping and paging comprise a DRX cycle. NB-IoT utilizes extended DRX (eDRX), which is an extension of DRX allowing the UE to experience much longer periods of inactivity (up to hours at a time). Further, UEs also have the option of entering Power Saving Mode (PSM). PSM allows the device to enter a deep sleep state for up to several days at a time. Upon waking from this deep sleep, the device must perform a Tracking Area Update (TAU), making the base station aware of its availability. These sleep modes are illustrated in figure 2. Another feature introduced by NB-IoT is coverage enhancement through repetition of messages. This is achieved through the time diversity gained by transmitting the same message at different points in time under different channel conditions. This technology, while increasing coverage, degrades the energy efficiency of the device by forcing it to transmit the same information multiple times.

B. LTE-M

LTE-M shares much of its philosophy with NB-IoT. LTE-M stems from legacy LTE, making some modifications to cater to the lower required data rates of much of the traffic that makes up Machine Type Communication (MTC). While the typical RF bandwidth of LTE is 20 MHz, LTE-M reduces this bandwidth requirement down to 1.4 MHz. In this sense, LTE-M provides a middle ground between legacy LTE and NB-IoT. While its bandwidth is not large enough for typical human user traffic, it is suitable for MTC at a higher data rate and lower delay than NB-IoT is capable of.

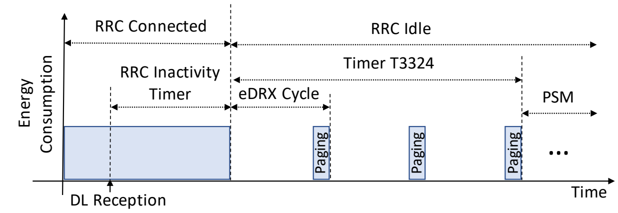

Fig. 2. "Timing diagram of NB-IoT energy components" inspired from [3]

LTE-M offers many improvements when compared with legacy LTE. Namely, it provides a *more efficient use of resources* through the ability to allocate fewer resources per transmission, it provides a *lower cost* due to a reduction in bandwidth, and it provides a *lower power consumption* due to a decrease in necessary transmission power.

III. Survey of Surveys

Popli et al. [4] give an in depth discussion on the workings of NB-IoT, with one section dedicated to the topic of energy efficiency. In this section, the authors classify energy efficiency techniques into five categories: sleep/wakeup techniques, cognitive radio techniques, routing techniques, data reduction techniques, and battery repletion techniques. The authors further break each section down into specific architectures/algorithms, and give a description and objective for each architecture/algorithm.

Wu et al. [5] review base station sleep mode techniques, and discusses the potential, feasibility, and applications of these techniques. The authors break down the energy efficiency problem in mobile networks into five categories. Similarly to [4], Wu identifies the categories of turning off components selectively, optimizing energy efficiency of the radio transmission process, and adopting renewable energy resources. Further, Wu introduces the categories of improving energy efficiency of hardware components and planning and deploying heterogeneous cells.

Buzzi et al. [6] provides a categorization of energy efficient techniques in 5G similar to that of [4] and [5], applied to 5G. Their breakdown of the problem considers four categories: Resource allocation, network planning and deployment (including locations of Base Stations (BSs), offloading techniques, and BS sleeping techniques), energy harvesting and transfer, and hardware solutions (including energy efficiency of hardware components and Radio Access (RA) technologies).

Abbas et al. [7] discuss energy conservation for wireless IoT. Their taxonomy is based on radio access technologies, i.e., Wireless Wide Area Network, Wireless Local Area Network, and Wireless Personal Area Network. They explore energy efficiency problems that exist in each technology. This work focuses on DRX/PSM, but also discusses overload control and energy efficient resource allocation.

Dash et al. [8] discuss and evaluates energy efficiency for 5G Wireless Sensor Networks (WSNs) and mobile ad-hoc networks. The authors break the problem down based on the layers of the IP stack: physical layer (e.g., energy efficient hardware components), MAC layer (e.g., scheduling to reduce collisions and retransmissions), Network layer (e.g., energy efficient routing), and Application layer (which varies greatly from application to application).

Shahraki et al. [9] discuss clustering in WSNs. The authors divide energy efficiency related to clustering into two broad categories: clustering techniques (including Cluster Head (CH) rotation, hierarchical clustering, and balancing clusters), and a broader IoT-specific section (including application-aware techniques and routing techniques in clustered architectures).

IV. Scheduling

The ways in which resources are utilized in a network is crucial to the power consumption and thus battery lifetime of that network. This includes the assignment of time-frequency resources to UEs and routing in the network.

A. Uplink

In [10], the authors propose a two-stage algorithm that optimizes scheduling within an NB-IoT network with a Quality of Service (QoS) constraint. The first stage determines the type and amount of resources each transmission will need. The second stage will order these transmissions based on delay constraints in order to minimize energy consumption while still meeting deadlines. Results show that energy consumption per UE remains the same as the number of UEs in the network increases. This is in contrast to other competing methods, whose energy consumption per UE increases slightly with an increasing number of UEs.

In [11], the authors proposes a genetic algorithm for resource scheduling in an LTE-M network. In Time Domain Packet Scheduling (TDPS), the available nodes in the network are selected using the Buffer Status Report (BSR) uploaded in the network, detailing the buffer size of each device. In Frequency Domain Packet Scheduling (FDPS), if there are more RBs than active nodes, a sub-optimal max-space algorithm is employed. Otherwise, a memetic algorithm is used, outputting the optimal set of M2M nodes. Further, post allocation, devices are assigned an "urgency weight" which are used to configure DRX parameters. This system was set up and tested using Network Simulator 3 (NS3). Results show that in general, as the number of nodes in the network increase, so does the energy savings, significantly outperforming round robin scheduling. However, in doing so packet loss increases.

B. Downlink

In [12], the authors proposes a scheduling algorithm to provide software updates to all devices in an IoT network. This algorithm optimizes the energy consumption of the distribution of the update with the constraint of a deadline by which the update must be in effect. Overall, this method increases energy efficiency by 12.8% compared to a solution based on IBM's CPLEX. In [13], a scheduling algorithm is introduced which reduces the energy consumption in the Downlink (DL) direction by prioritizing the transmission of messages that will travel longer distances and/or more hops before reaching their destinations. Since these messages will require more energy, the prioritization of them will reduce overall energy consumption. The introduction of this algorithm was shown to provide a 36% increase in energy efficiency when compared with no scheduling optimization in place.

In [14], two similar algorithms, referred to as global and light, are proposed, designed for the transmission of bulk data in a LoRa-WAN based network. Slots of variable size are assigned to the transmission of data based on that transmission's Spreading Factor (SF). The algorithm goes through the available transmissions in the network in decreasing order based on their SF. It will then assign each transmission to the earliest possible slot, while still considering the required radio duty cycle. This allows for a quicker data collection time, while simultaneously reducing the energy consumption in the network. Through simulation, both algorithms are compared with LoRaWAN, which they outperform in terms of energy efficiency by up to 250%.

C. Both Uplink and Downlink

In [15], the authors discuss Uplink (UL) and DL scheduling in NB-IoT. They also evaluate tradeoffs in Key Performance Indicators (KPIs) as a result of scheduling, including the tradeoff between latency and energy consumption.

In [16], a genetic algorithm is proposed to improve the energy efficiency of routing in WSNs, which often utilize C-IoT technologies. This algorithm uses a sleep-wake up mechanism that selects some nodes in the network to sleep, thus reducing their energy consumption. The algorithm then only searches through the active nodes in the network to search for solutions. Once one of the active nodes begins running out of power, one of the sleeping nodes is turned into an active node. A local search technique is also employed. With each iteration of the algorithm, the search expands slightly, exploring neighboring options. The algorithm stops once set constraints are met. Simulation results show an energy efficiency improvement of around 33% compared with competing algorithms. Further, the developed algorithm also improves upon the number of successful packet transmissions as well as the total delay.

In [17], the authors propose a scheduling algorithm for WSNs based on traffic priority. The algorithm computes this priority using the required data rate, minimum delay, transmitting power, remaining energy, and remaining buffer size of devices for each transmission. The base station then allocates collision-free time slots for transmissions based on this priority. The duty cycle of each device is also altered in this algorithm based on the above factors. The algorithm was simulated and compared with Energy Efficient Context Aware Traffic Scheduling (EE-CATS) [18], which it outperforms in terms of energy efficiency, packet drop rate, and throughput.

V. Data Processing

By altering where, when, and how calculations are performed in C-IoT, energy can be saved. This can be achieved using data reduction techniques, energy efficient circuitry for computation, and task offloading (e.g. to the cloud).

A. Minimizing Transmitted Data

In [19], the authors develop a delta compression coding scheme for temporally correlated data that can increase the energy efficiency of compression by up to 85%. This coding scheme also increases the compression ratio and decreases memory requirements for compression. However, delta compression relies on the data being temporally correlated, therefore real time applications are not suitable for this method. In [20], the authors propose an algorithm that prunes a data structure, removing insignificant value tags within annotated data that can be recovered later. This process takes place between the gateway and cloud, and reduces the amount of data to be transmitted. In doing so, transmission energy is reduced and the message is still entirely recoverable.

B. Approximate Circuitry

In [21], the authors employ inexact arithmetic circuits to test the tradeoff between energy efficiency and accuracy. Results show that if data is processed in this manner, there can be an improvement of up to 80.4% in energy efficiency, with a tradeoff of 1.34 dB in SNR at the output. In [22], the authors propose a scheme in which the UE decides how much energy it dedicates to computation based on the battery percentage of the device. Simulation results show that maximum energy consumption of the device is reduced by 38%, and the average energy consumption in the network is decreased by 6%.

C. Task Offloading

In 5G networks, much of the data processing can be done through cloud computing, rather than physically in the UEs. In [23], the authors discuss fog computing and cloud computing in the context of 5G networks. Fog computing provides an intermediate step in the hierarchy of computing between the centralized cloud and the widely distributed, low cost UEs. In introducing this, local resources can be used for some computation instead of the potentially farther away cloud, which increases energy efficiency in the network. In [24], the authors propose a method of offloading computational tasks to other, more capable nodes in the network, such as edge nodes, distributed units, and central units. This method distributes the computational load in a way that the computation will take less energy overall. Experimental results show that relative to competing algorithms, the developed algorithm saves energy when considering a larger number of computational tasks, increasing with an increasing number of tasks.

In [25], the author designs a task allocation algorithm for a distributed m2m network. In this algorithm, the base station first takes a large job and breaks it into tasks, then further into sub-tasks. Each task will be forwarded to a group of neighboring nodes that are combined into clusters. In this cluster, a CH is determined by which node in the cluster has the greatest residual energy. This CH is responsible for forwarding these sub-tasks to other nodes in the cluster, as well as aggregating data and transmitting it back to the base station. A simulation was conducted in MATLAB that shows that while devices were depleting their available energy quicker, tasks were also being completed at a much quicker rate, thus resulting in a net increase in terms of efficiency.

In [26], an architecture is proposed in which data processing can be conducted within a WSN. This architecture consists of three layers. The data clustering layer allows the sensors in the network to aggregate data and transmit accurate data efficiently to the cluster head. In the data compression layer, data is compressed at the cluster head prior to transmission to the gateway. In the data mining level, gateway nodes accomplish tasks such as translation, filtering, and integration of data in order to detect more complex events. This data is then sent to users as knowledge. This architecture was simulated through MATLAB and was found to save approximately 40% more energy when compared to a centralized approach.

In [27], the authors propose a Delay-Based Workload Allocation algorithm, which decides whether a task should take place in a local edge server, a neighboring edge server, or in the cloud. Based on the required delay for the task, the algorithm balances the delay-energy tradeoff such that this delay requirement is met while energy consumption is reduced.

Both task offloading and approximate circuitry are combined in [28]. In the authors' proposed algorithm, tasks can be executed in four different manners: In End Device (ED) exactly, in ED approximately, in Fog Node (FN) exactly, or in FN approximately. The authors formulate this optimization as a linear programming problem that considers the average processing power of EDs and FNs, the transmission cost required to offload computation from ED to FN, and an error threshold predefined by the user. Simulation is conducted in MATLAB, where the network lifetime was shown to have an approximate 200% improvement relative to a comparable algorithm. However, the developed algorithm has a greater complexity when compared with the same comparable algorithm. This complexity increases more significantly as the numbers of end devices and tasks increase.

VI. Sleep Modes

By allowing a device to enter periods of time where RF activity is either reduced or stopped entirely, the device can save energy it would usually spend monitoring networking channels. This can be accomplished through two means in UEs, namely DRX and PSM. In [29], the authors introduce both of these mechanisms, provide an analytical model for each, and evaluate the performance of both mechanisms through their implementation in NS3 using the NB-IoT protocol.

A. DRX

In [33], the DRX mechanism is evaluated through a developed cross-layer analytical model with traffic distributed according to a Poisson process. Results show that the introduction of the DRX mechanism results in a considerable improvement (up to three times) in the energy efficiency of the device. Further, results show that for given DRX timers, there is a certain traffic load at which the energy efficiency improvement of the mechanism is optimum. This illustrates

TABLE I
COMPARISON OF ENERGY-SAVING TECHNOLOGIES

Ref.	Brief Description	Energy-Saving Method	Methodology	Tradeoffs
[12]	Scheduling over-the-air software updates with deadlines	DL scheduling	Heuristic algorithm	Delay
[13]	Scheduling prioritizing packets going far and over many hops	DL scheduling	Dijkstra + queuing theory	NA
[10]	Two-stage scheduling for energy efficiency under QoS constraints	UL scheduling	NA	Delay + reliability
[11]	Optimizes energy under QoS constraint in FDPS	UL scheduling	Genetic algorithm	Packet loss
[16]	Genetic algorithm using local search technique to calculate routes	P2P scheduling	Genetic algorithm	NA
[17]	Priority-based and collision-free scheduling algorithm	UL + DL scheduling	Dynamic programming	Delay + throughput
[14]	Algorithm for bulk data transmission in LoRa network	UL + DL scheduling	Dynamic programming	Complexity
[24]	Balances computation between end-nodes and central units	Task offloading	Evolutionary algorithm	Complexity
[28]	Algorithm deciding where and how a computation will take place	Approx. circuitry + task offloading	Linear programming	Complexity
[25]	Tasks are broken into subtasks and distributed to network devices	Task offloading	Divide and Conquer	NA
[26]	An in-network layered data processing architecture is proposed	Task offloading	Hierarchical architecture	Scalability
[27]	Balances delay-energy tradeoff of task offloading	Task offloading	Queueing theory	Delay
[19]	Delta compression scheme for temporally correlated data	Data minimization	Coding theory	Accuracy
[20]	Insignificant data is removed from data structures at the gateway	Task offloading	Ontology-based algorithm	NA
[22]	Energy dedicated to computation is based on battery status	Approx. circuitry	Linear programming	Computation + communication
[30]	Selects one of four sleep levels based on coverage and traffic load	BS sleeping	Linear programming	Delay
[31]	"Quick Sleeping Indicator" allows a device to skip a paging interval	DRX	NA	NA
[32]	RL is used to improve the latency-energy tradeoff in DRX	DRX	Machine learning	Delay

the importance of choosing DRX timers according to traffic load to achieve the best energy efficiency and delay results.

In [32], the authors propose an actor-critic algorithm to improve the latency-energy tradeoff that exists in DRX. The authors consider a modified DRX mechanism consisting of four states: continuous reception, on duration of DRX cycle, off duration of DRX cycle, and Radio Resource Control (RRC) Idle. The algorithm learns over time through the modification of the timers that facilitate state transitions (e.g., on duration of DRX cycle). The authors evaluate the proposed algorithm using MATLAB, and find that the proposed algorithm outperforms standard eDRX in terms of energy efficiency by approximately 300%. However, the average delay of the actor-critic algorithm is much greater at approximately 280 ms, compared with conventional DRX at 50 ms.

In [31], the authors discuss a "quick sleeping" mechanism for DRX, consisting of a Quick Sleep Indicator (QSI) that is sent to the UE after it wakes up from DRX sleep. The QSI can be deployed in either the Physical Broadcast Channel (PBCH) or in the Primary Synchronization Signal (PSS)/Secondary Synchronization Signal (SSS). This message will tell the UE to either go back to sleep, or to stay awake. Only if the UE is told to stay awake will it continue on to decode paging information. Compared to conventional DRX, where the UE stays awake to decode paging information 100% of the time, this mechanism gives a chance for the UE to go back to sleep without wasting energy staying awake and decoding paging information. A simulation was conducted using MATLAB, and it was found that the introduction of the quick sleeping mechanism brought with it a 45% reduction in energy consumption.

B. PSM

In [34], the authors use a genetic algorithm to determine the optimal PSM timer duration. Results show that through the optimization of the PSM timer duration, the device can save a considerable amount of energy while still meeting latency constraints. Further, the authors show through simulation that beyond a certain timer threshold, increasing the PSM timer further has little impact on the power saving factor.

In [35], the authors model PSM through a semi-Markov process consisting of four states: PSM, idle, RA, and transmitting. The developed model considers periodic UL reporting, where the UE generates UL traffic at regular intervals. Results are consistent with [34], showing that continuing to increase the PSM beyond certain values will only degrade network performance in the form of latency, while having little effect on the power saving factor.

C. Base Station

In [30], the authors propose a framework that minimizes power consumed by a small cell base station while maintaining an adequate QoS for users in the cell. They do so by scheduling each cell's state (on, waiting, deep sleep, or off) based on coverage areas and traffic load. In [36], the authors evaluate and model Advanced Sleeping Mode (ASM) in 5G networks. In ASM, there are 4 different modes of sleep, varying in amount of time spent sleeping and in the amount of power saved by shutting off some components.

In [37], the authors develop a joint algorithm that considers both clustering and sleeping of BSs. The inner problem, clustering, is solved using an optimal algorithm of polynomial complexity. The outer problem, sleeping, is solved through either a greedy algorithm that calculates a nearly optimal solution, or an optimal algorithm whose search space is less extensive than an exhaustive search, but more extensive than the greedy algorithm. Through simulation, the authors verify that the developed joint algorithm can achieve energy savings of up to 80% compared with no clustering/sleeping algorithm.

VII. CONCLUSION

In this work, we presented the state of the art in energy-saving for C-IoT technologies, and the resulting network performance tradeoffs. The available literature was categorized into three broad categories: scheduling, data processing, and sleep modes. Works in each of these areas were discussed and analyzed. Table I summarizes each work and provides the tradeoff resulting from decreasing energy consumption.

We recommend that future research in this area more systematically quantifies tradeoffs between energy and performance. In doing so, it will be easier to identify how each technology will perform in different scenarios and across various energy or performance constraints. With this knowledge, comparisons between technologies will become clearer, allowing a more well informed selection of the technology based on given requirements. Additionally, it is recommended that more studies be conducted using real hardware components. Currently, the majority of results are found using simulation and mathematical models, which sacrifice reality for feasibility. If more studies are conducted using hardware, we will get a more realistic idea of how implementing new energy-saving technologies will impact the energy-performance tradeoff.

REFERENCES

[1] L. Goasduff, "Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will be in Use in 2020," Gartner, Tech. Rep., Aug 2019, tinyurl.com/342xyupn.

[2] M. Kanj, V. Savaux, and M. Le Guen, "A Tutorial on NB-IoT Physical Layer Design," *IEEE Commun. Surveys Tuts*, vol. 22, no. 4, pp. 2408–2446, 2020.

[3] S. Oh, K. Jung, M. Bae, and J. Shin, "Performance analysis for the battery consumption of the 3GPP NB-IoT device," in *Int. Conf. on Inf. and Commun. Technol. Convergence (ICTC)*, 2017, pp. 981–983.

[4] S. Popli, R. K. Jha, and S. Jain, "A Survey on Energy Efficient Narrowband Internet of Things (NBIoT): Architecture, Application and Challenges," *IEEE Access*, vol. 7, pp. 16 739–16 776, 2019.

[5] J. Wu, Y. Zhang, M. Zukerman, and E. K. Yung, "Energy-Efficient Base-Stations Sleep-Mode Techniques in Green Cellular Networks: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 803–826, 2015.

[6] S. Buzzi, C. I, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, "A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 697–709, 2016.

[7] Z. Abbas and W. Yoon, "A Survey on Energy Conserving Mechanisms for the Internet of Things: Wireless Networking Aspects," *Sensors*, vol. 15, pp. 24 848–24 847, 2015.

[8] L. Dash and M. Khuntia, "Energy efficient techniques for 5G mobile networks in WSN: A Survey," in *2020 Int. Conf. on Comput. Sci., Eng. and Appl. (ICCSEA)*, March 2020, pp. 1–5.

[9] A. Shahraki, A. Taherkordi, Haugen, and F. Eliassen, "A Survey and Future Directions on Clustering: From WSNs to IoT and Modern Networking Paradigms," *IEEE Trans. Netw. Service Manag*, 2020.

[10] P. Liu, K. Wu, J. Liang, J. Chen, and Y. Tseng, "Energy-Efficient Uplink Scheduling for Ultra-Reliable Communications in NB-IoT Networks," in *2018 IEEE 29th Annual Int. Symp. on Perso., Indoor and Mobile Radio Commun. (PIMRC)*, 2018, pp. 1–5.

[11] S. Dawaliby, A. Bradai, Y. Pousset, and C. Chatellier, "Joint Energy and QoS-Aware Memetic-Based Scheduling for M2M Communications in LTE-M," *IEEE Trans. Emerg. Topics Comput. Intell*, vol. 3, no. 3, pp. 217–229, 2019.

[12] N. H. Bui, C. Pham, K. K. Nguyen, and M. Cheriet, "Energy Efficient Scheduling for Networked IoT Device Software Update," in *2019 15th Int. Conf. on Netw. and Service Manage. (CNSM)*, pp. 1–5.

[13] L. Farhan, A. E. Alissa, S. T. Shukur, M. Hammoudeh, and R. Kharel, "An energy efficient long hop (LH) first scheduling algorithm for scalable Internet of Things (IoT) networks," in *2017 Eleventh Int. Conf. on Sens. Technol. (ICST)*, pp. 1–6.

[14] D. Zorbas, K. Q. Abdelfadeel, V. Cionca, D. Pesch, and B. O'Flynn, "Offline Scheduling Algorithms for Time-Slotted LoRa-based Bulk Data Transmission," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 949–954.

[15] A. Azari, Stefanović, P. Popovski, and C. Cavdar, "On the Latency-Energy Performance of NB-IoT Systems in Providing Wide-Area IoT Connectivity," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 1, pp. 57–68, 2020.

[16] A. S. Hampiholi and B. P. Vijaya Kumar, "Efficient routing protocol in IoT using modified Genetic algorithm and its comparison with existing protocols," in *2018 3rd Int. Conf. on Circuits, Control, Commun. and Comput. (I4C)*, Oct 2018, pp. 1–5.

[17] K. Kavitha and G. Suseendran, "Priority based Adaptive Scheduling Algorithm for IoT Sensor Systems," in *2019 Int. Conf. on Automat., Comput. and Technol. Manage. (ICACTM)*, 2019, pp. 361–366.

[18] B. Afzal, S. A. Alvi, G. A. Shah, and W. Mahmood, "Energy efficient context aware traffic scheduling for iot applications," *Ad Hoc Networks*, vol. 62, pp. 101–115, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870517300823

[19] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient IoT solutions," in *2017 25th Telecommun. Forum*, pp. 1–4.

[20] Y. Xu and T. Kishi, "An Ontology-Based IoT Communication Data Reduction Method," in *2018 9th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, pp. 321–325.

[21] M. Osta, A. Ibrahim, H. Chible, and M. Valle, "Inexact Arithmetic Circuits for Energy Efficient IoT Sensors Data Processing," in *2018 IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, pp. 1–4.

[22] W. S. Ravindra and P. Devi, "Energy balancing between computation and communication in IoT edge devices using data offloading schemes," in *2018 Second Int. Conf. on Inventive Commun. and Comput.l Technol. (ICICCT)*, pp. 22–25.

[23] S. Kitanov and T. Janevski, "Energy efficiency of Fog Computing and Networking services in 5G networks," in *IEEE EUROCON 2017 -17th Int. Conf. on Smart Technol.*, 2017, pp. 491–494.

[24] X. Liu, X. Xu, Y. Yuan, X. Zhang, and W. Dou, "Energy-Efficient Computation Offloading with Privacy Preservation for Edge Computing-Enabled 5G Networks," in *2019 Int. Conf. on Internet of Things (iThings) and IEEE Green Comput. and Commun. (GreenCom) and IEEE Cyber, Physical and Social Comput. (CPSCom) and IEEE Smart Data (SmartData)*, 2019, pp. 176–181.

[25] S. A. Aboalnaser, "Energy–Aware Task Allocation Algorithm Based on Transitive Cluster-Head Selection for IoT Networks," in *2019 12th Int. Conf. on Develop. in eSystems Eng. (DeSE)*, 2019, pp. 176–179.

[26] W. He, S. Yang, L. Yang, and P. Li, "In-network data processing architecture for energy efficient wireless sensor networks," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 299–304.

[27] M. Guo, L. Li, and Q. Guan, "Energy-Efficient and Delay-Guaranteed Workload Allocation in IoT-Edge-Cloud Computing Systems," *IEEE Access*, vol. 7, pp. 78 685–78 697, 2019.

[28] W. Yu, A. Najafi, Y. Nevarez, Y. Huang, and A. Garcia-Ortiz, "TAAC: Task Allocation Meets Approximate Computing for Internet of Things," in *2020 IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, 2020, pp. 1–5.

[29] A. K. Sultania, P. Zand, C. Blondia, and J. Famaey, "Energy Modeling and Evaluation of NB-IoT with PSM and eDRX," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–7.

[30] M. S. A. Shuvo, A. R. Munna, T. Adhikary, and M. A. Razzaque, "An Energy-Efficient Scheduling of Heterogeneous Network Cells in 5G," in *2019 Int. Conf. on Sustain. Technol. for Industry 4.0 (STI)*, pp. 1–6.

[31] N. M. Balasubramanya, L. Lampe, G. Vos, and S. Bennett, "DRX With Quick Sleeping: A Novel Mechanism for Energy-Efficient IoT Using LTE/LTE-A," *IEEE IoT Journal*, vol. 3, no. 3, pp. 398–407, 2016.

[32] J. Zhou, G. Feng, T. P. Yum, M. Yan, and S. Qin, "Online Learning-Based Discontinuous Reception (DRX) for Machine-Type Communications," *IEEE IoT Journal*, vol. 6, no. 3, pp. 5550–5561, June 2019.

[33] B. Cai, Y. Chen, and I. Darwazeh, "Analyzing Energy Efficiency for IoT Devices with DRX Capability and Poisson Arrivals," in *2019 26th Int. Conf. on Telecommun. (ICT)*, pp. 254–259.

[34] K. Liu, G. Cui, Q. Li, S. Zhang, W. Wang, and X. Li, "An Optimal PSM Duration Dalculation Algorithm For NB-IoT," in *2019 IEEE 5th Int. Conf. on Comput. and Commun. (ICCC)*, pp. 447–452.

[35] H. Bello, X. Jian, Y. Wei, and M. Chen, "Energy-Delay Evaluation and Optimization for NB-IoT PSM With Periodic Uplink Reporting," *IEEE Access*, vol. 7, pp. 3074–3081, 2019.

[36] F. E. Salem, T. Chahed, E. Altman, A. Gati, and Z. Altman, "Optimal Policies of Advanced Sleep Modes for Energy-Efficient 5G networks," in *2019 IEEE 18th Int. Symp. on Netw. Comput. and Appl.*, pp. 1–7.

[37] J. Kim, H. Lee, and S. Chong, "Traffic-Aware Energy-Saving Base Station Sleeping and Clustering in Cooperative Networks," *IEEE Trans. Wireless Commun*, vol. 17, no. 2, pp. 1173–1186, 2018.