A Sub-1 GHz Wireless Sensor Network Concentrator Using Multicollectors with Load Balancing for Improved Capacity and Performance

Ricardo Arjona, Craig Lee, Miguel Razo, Marco Tacca and Andrea Fumagalli
The University of Texas at Dallas, Richardson, TX, USA
Universidad Tecnológica de Bolívar, Cartagena, Colombia
[rja150230, craig.lee, mrazo, mtacca, andreaf]@utdallas.edu

Kumaran Vijayasankar
Texas Instruments
Dallas, TX, USA
kumaran@ti.com

***Abstract*—The exponential growth of IoT end devices creates the necessity for cost-effective solutions to further increase the capacity of IEEE802.15.4g-based wireless sensor networks (WSNs). For this reason, the authors present a wireless sensor network concentrator (WSNC) that integrates multiple collocated collectors, each of them hosting an independent WSN on a unique frequency channel. A load balancing algorithm is implemented at the WSNC to uniformly distribute the number of aggregated sensor nodes across the available collectors. The WSNC is implemented using a BeagleBone board acting as the Network Concentrator (NC) whereas collectors and sensor nodes realizing the WSNs are built using the TI CC13X0 LaunchPads. The system is assessed using a testbed consisting of one NC with up to four collocated collectors and fifty sensor nodes. The performance evaluation is carried out under race conditions in the WSNs to emulate high dense networks with different network sizes and channel gaps. The experimental results show that the multi-collector system with load balancing proportionally scales the capacity of the network, increases the packet delivery ratio, and reduces the energy consumption of the IoT end devices.**

***Index Terms*—Sub-1 GHz, IEEE802.15.4g, multi-collector, load balancing, wireless sensor network concentrator, IoT**

I. Introduction

Wireless Sensor Networks – WSNs are a major player in the massive deployment of end devices in the Internet of Things – IoT. By wirelessly interconnecting control systems with sensors and actuators to interact with the physical world, WSNs simplify the infrastructure for gathering data thus substantially reducing solution costs. This IoT growth is in part supported by a variety of wireless protocol standards addressing different types of application requirements ranging from energy consumption, latency, connectivity range, bandwidth, and spectrum allocation among others [1]- [2]. The focus of this paper is centered on the Sub-1 GHz IEEE802.15.4g-based Low Rate Wireless Personal Area Networks (LR-WPAN), which is a star topology network conceived to support a large number of devices deployed over extensive geographical areas with minimal infrastructure.

The work of R. Arjona was co-sponsored by Fulbright, the Colombian Ministry of Education, and the Universidad Tecnológica de Bolívar under Grant G-1-12218, and Texas Instruments, Inc. This work is supported in part by NSF grants CNS-1405405, CNS-1409849, ACI-1541461, CNS-1531039, and CNS-1956357.

The IEEE802.15.4g is designed to realize the Smart Utility Networks – SUN where connectivity of at least 1,000 direct neighbors in a dense urban network is expected [6] using a single Medium Access Control – MAC with three alternative physical layers (FSK, OFDM and O-QPSK) over a range of unlicensed frequencies to meet different regulatory domains [2]. Under this protocol, multiple sensor nodes connect to a collector node over a wireless link to form a star topology network. By supporting multiple data rates starting from 5 kbps and up to 400 kbps, wherein 50 kbps is the most common setting, this technology is suitable for a number of potential applications ranging from smart metering to smart agriculture in both urban and rural environments.

A common requirement in the family of IEEE 802.15.4-based wireless technologies is to reduce the power consumption at the end devices and maximize their battery lifespan. However, the exponential growth of IoT devices combined with the use of unlicensed spectrum is making LR-WPAN more susceptible to congestion and transmission interference whereby the classic IoT assumption of low duty cycle is no longer applicable [3]. These factors contribute to increased energy consumption at the end devices and limited system scalability as described next.

In dense WSNs, the increased number of sensor nodes accessing the radio channels causes race conditions to be more likely to occur. In such environments, sensor nodes experience an increased radio usage associated to the increased channel contention, packet loss, retransmission attempts, and channel access failures that all combined translate into additional power consumption at the end devices. In terms of interference, studies have identified the Co-Channel Interference – CCI as a major performance degrading factor in coexisting IEEE802.15.4g physical layers (homogeneous) and among dissimilar systems (heterogeneous) affecting each other, such as 802.11b/g/n [4] and Lora [5] beside IEEE802.15.4g network. Despite the provisions included in the specifications to deal with homogeneous systems, such as the Multi-PHY Management – MPM protocol [2] [6], the surge of IoT devices is posing these problems within the same physical layer in a IEEE802.15.4g-based WSN, which has not been

experimentally documented yet. On the network scalability, recent research activities have focused on technologies other than IEEE802.15.4g. For example, the analysis of large-scale LoRaWAN networks in [7] shows the detrimental impact the downstream traffic has on the delivery ratio of confirmed upstream traffic when the network grows from hundreds to thousands of end devices in simulations based on NS-3. The study reveals that increasing gateway density can ameliorate but not eliminate this effect due to duty cycle requirements. Similarly, the work in [8] proposes the use of an additional base station to improve scalability of the long range SigFox communication protocol.

Similar solutions have been proposed mostly for LR-WPAN in the 2.4 GHz spectrum without accounting for Sub-1 GHz IEEE802.15.4g-based WSNs. The work in [9] introduces a multiradio gateway scheduling algorithm for 2.4 GHz IEEE802.15.4e-based WSN to overcome the problem of gateway saturation (bottleneck) which improves the duty cycle and capacity in Time Synchronize Channel Hopping Networks – TSCH for Industrial IoT – IIoT. In [10] [11], a LoRaWAN gateway is reported — consisting of a Raspberry Pi 3B+ and an iC880A concentrator LoRa Gateway board — which supports up to 8 LoRa packets simultaneously having different spreading factors and different receiving channels. Likewise, the authors in [12] conduct an experimental evaluation of a LoraWAN dual-gateway wherein adding a second gateway in the uplink dramatically increases the delivery rate and reduces the packet loss when operating in heavy uplink traffic conditions.

In this paper the authors address the aforementioned problems — in which the collector node becomes the bottleneck of the network — by introducing a WSNC consisting of collocated *collectors* that are managed by a *Network Concentrator – NC* to create multiple, independent, and coexisting WSNs over different frequency channels. This architecture expands the capabilities of the single collector platform presented in [13] — that supports a maximum of 50 sensor nodes due to memory constraints at the collector node — by scaling up the network size as a function of the number of deployed collectors. The WSNC enables multiple *sensor nodes* to concurrently send data through any of the available collectors. More specifically, the NC is implemented on a BeagleBone Black Wireless device [14] running an embedded Linux distribution while collectors and sensors nodes realizing the WSNs are built on a Texas Instruments CC1310 Wireless Microcontroller Unit – WMCU [15] using the SimpleLink TI 15.4 Stack Protocol [16]. The second contribution of this paper is a load balancing algorithm that exploits the capacity provided by the WSNC to concurrently reduce race conditions, packet collisions, and the energy consumption at the sensor nodes. The algorithm operates in conjunction with a handover mechanism also implemented by the authors that enables a sensor node to hop from one collector to another in order to reach the desired load per collector. The third contribution of this paper is the extensive evaluation of the multicollector system relative to its single collector version for the Multi-Rate Fequency Shift Keying (MR-FSK) operating mode # 1. The assessment is carried out using a Sub-1 GHz WSNC testbed consisting of 4 collectors and 50 sensor nodes devised to emulate large and dense networks with different network sizes, channel gaps and number of collectors. To the best of the authors' knowledge this is the first IEEE802.15.4g-based multi-collector system experimentally evaluated using a testbed with a large number of sensor nodes. The experimental results show that the WSNC increases its capacity and reduces its packet error rate while concurrently reducing the energy consumption at the sensor nodes through the use of multiple collectors in large and dense WSNs.

II. The Wireless Sensor Network Concentrator

The WSNC uses a Network Concentrator – NC to integrate several collocated collectors and form a single entity to proportionally expand the capacity and performance of traditional single collector IEEE802.15.4-based star topology WSNs. As depicted in Fig. 1, the WSNC system comprises three network components: the NC, the collectors and the sensors nodes. The NC interacts with each collector over a point serial communication using UART as the underlying physical interface. Each collector realizes an independent WSN that is managed by the NC over a unique frequency channel. Concurrent network access of the sensor nodes is achieved through the muliple collectors. This design enables end-to-end connectivity between the NC and the sensor nodes via the collectors. The rest of this section describes each component of the WSNC system along with the features implemented.

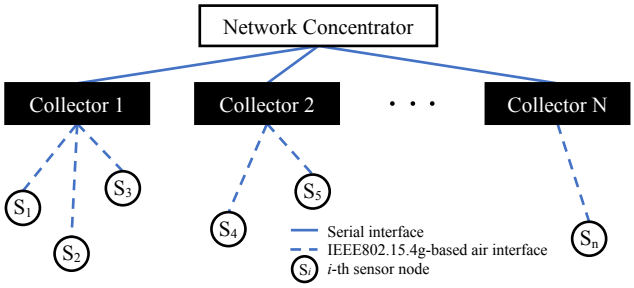

Fig. 1. WSNC architecture diagram with N collectors.

A. The Network Concentrator

The NC is the brain of the WSNC system and is implemented using a Beaglebone Black Wireless – BBW microcomputer running a Linux embedded distribution as the operating system. Collectors are attached to the BBW using one of its four built-in UART serial ports. The key functionalities of the NC are implemented in user-space. Through this application, the following three basic tasks are implemented:

- *Gateway*: to provide connectivity to external networks such as a remote Cloud server, through the provisioned LTE connection over the WNC M14A2A LTE Cat 1 modem.
- *Network management*: to locally or remotely configure, operate, and manage every WSN hosted by the collectors.

- *Concentrator*: to seamlessly aggregate the data traffic received from all the sensors through their respective collectors. The data can then be processed either locally in the NC or remotely in the Cloud depending on the application requirements.

B. The Collectors and Sensor Nodes

Collectors and sensor nodes realizing the independent WSNs managed by the NC are implemented using a number of TI CC1310 LaunchPads. The CC1310 is a WMCU designed for low-power and long-range wireless IoT applications in the Sub-1 GHz spectrum. Each collector attached to the NC uses the onboard XDC100 debug probe of the TI CC1310 LaunchPad, which provides serial-over-USB communication with the WMCU.

The SimpleLink TI 15.4 Stack Protocol — a proprietary solution from TI based on the IEEE802.15.4g standard — is the underlying protocol stack on top of which additional features are added to both the collector and sensor node's protocol stacks. At the collector side, such features include the serial interface to the NC for data transport, MAC event notification and configuration (e.g., status change, association, disassociation notifications, and MAC PIB attribute modification), and an enriched set of MAC configuration over-the-air messages to manage sensors in a more granular manner (e.g., transmit power, frame control, disassociation, and handover mechanisms among others). Similarly, at the sensor side the interface to communicate with the NC through the collectors is implemented. Through this interface the NC can modify the application parameters received by the sensor nodes after association, MAC PIB attributes, or to instruct a sensor to interact with the physical world. The software development and relative loading into the WMCU is done using the Code Composer Studio IDE.

Although, all the implemented tasks are of paramount importance to any IoT application, the focus of this paper is on those tasks that enable flexibility and network efficiency. In particular, we dedicate the next section to describe the key network management processes such as the handover procedure, and load balancing algorithm.

III. The Handover Procedure

The handover procedure is the process by which a wireless sensor node is instructed to switch from its parent collector to a neighbor collector, referred to as *target collector*. As depicted in Fig. 2, this procedure is initiated by the NC after sending a handover request message to a sensor node through its parent collector. The request is delivered to the sensor node through indirect messaging containing the channel frequency and address of the target collector along with the address of the sensor node subject to handover. At the sensor node, the message is processed by triggering the disassociation mechanism to leave its current parent collector followed by an association request to the target collector. The handover procedure is marked as complete when the sensor node is finally associated with the target collector and registered at the NC. It is important to notice from the above description that power measurements from neighboring collectors are not part of this procedure in order to avoid the energy consumption at the sensor node that would result from performing this task. Hence, the implemented procedure is a simplified version of the classical handover mechanism present in traditional wireless mobile networks. Accounting for the handover procedure total time that is at least $POLLING\_INTERVAL\_TIME$[1] + $DISASSOCIATION\_TIME$[2] + $ASSOCIATION\_TIME$[3], a timer at the NC is used to abort this procedure on timeout if unsuccessful and to clear all the associated states of the sensor node in question.

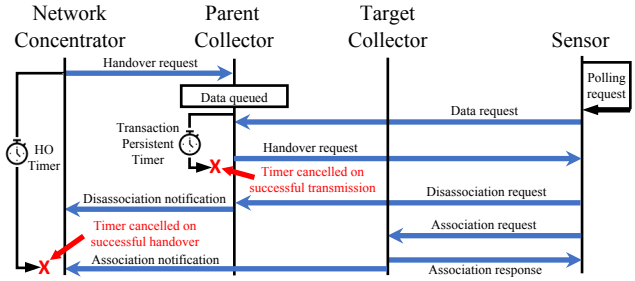

Fig. 2. Simplified handover sequence – MAC acknowledgments are not shown.

A. The Load Balancing Algorithm

The NC is equipped with a load balancing algorithm. The primary objectives of this algorithm are to minimize the energy consumption at the sensor nodes and to improve the network performance. In the WSNC system, sensor nodes join the network after scanning the 129 channels available in the 915 MHz band to find an available collector to associate with. Since lower channels offer better RF propagation, sensor nodes mostly attach to the lowest channels. This procedure leads to an uneven use of the radio channels potentially creating unnecessary congestion at the collectors that operate at the lowest channels while other collectors are underutilized. This effect increases the collision probability and the number of retransmissions triggered in the networks hosted by the congested collectors. Hence, the energy consumed by the sensor node is negatively impacted due to the increased radio usage that shortens its battery lifetime.

We use the notation in Table. I, to describe the implemented load balancing algorithm (see Alg. 1) which mitigates the aforementioned problem by uniformly distributing the total number of sensors N_S across all the available collectors. The algorithm is executed at the NC whenever a new sensor node joins the network. A simple sensor head count is used to balance the number of sensor nodes per collector whereby no quality of service indication is taken into consideration when moving sensors between collectors. The algorithm involves

[1]The time interval used by a sensor node to periodically poll the collector.
[2]The minimum amount of time required to disassociate from the parent collector.
[3]The minimum amount of time required to associate with a collector.

three basic steps that are described in the remainder of this section.

TABLE I
SYMBOL NOTATION

Symbol	Description
N_S	Total number of sensor nodes in the network
N_C	Total number of parent collectors in the network
p_i	i-th parent collector node
s_j	j-th sensor node in the aggregated network
$ch(p_i)$	Set of child sensor nodes of p_i
$C(p_i)$	Maximum capacity of p_i
$P(r)$	Ordered subset of parent collectors at data rate r
$S(r)$	Subset of sensor nodes at data rate r
$ho(r)$	Set of sensor nodes at data rate r to perform handover
$rssi_j$	RSSI of j-th sensor node at its parent collector
chg	Channel separation between parent collectors

1) Determining Collectors Capacity: The maximum capacity for each collector $C(p_i)$ with $1 \leq i \leq N_C$ is calculated in two steps. First, the subset $P(r)$ of parent collectors at a given data rate r is created from the set of attached collectors to the NC $\{p_1, p_2, \ldots, p_{N_C}\}$. Consequently, the children of each $p_i \in P(r)$ form together the subset $S(r)$ of sensor nodes operating at the data rate r in the aggregated network. Second, if $|S(r)|\%|P(r)| = 0$ then all $p_i \in P(r)$ get the same maximum capacity $C(p_i) = |S(r)|/|P(r)|$, otherwise, the subset $P(r)$ is sorted by ascending order based on their channel indexes, and the remainder $|S(r)|\%|P(r)|$ is round robin distributed over the sorted set $|P(r)|$ as described in Alg. 1. This sorting ensures that the collectors that operate at lower frequencies have one extra sensor node compared to the collectors that operate at higher frequencies.

2) Handover Sensor Node Selection: The completion of a handover procedure is subject to the availability of a reliable over-the-air communication between the parties, whereby handover failures are mainly driven by temporary poor channel conditions. To minimize the handover failures, sensor nodes are selected based on the highest received signal strength indicator – RSSI to increase the probability of success. Here, we use the RSSI — measured by the receiver of the collector on a per packet basis — as an indicator of the uplink channel quality, that is, the higher the RSSI the better the link quality. The uplink radio performance at the target collector is expected to be similar to the current parent collector's under the assumption that these two collectors are collocated at the NC and the distances from sensor-to-parent and sensor-to-target collectors are mainly the same.

Therefore, once the maximum capacities $C(p_i)$ have been set, the next step is to select the sensor nodes that need to be moved from an overloaded collector to an underloaded collector. As depicted in Alg. 2, the number of sensor nodes that must be removed for a collector is given by $C(p_i) - |ch(p_i)|$, subject to $|ch(p_i)| > C(p_i)$. The sensor nodes in $ch(p_i)$ are sorted by decreasing RSSI, and the first $C(p_i) - |ch(p_i)|$ in this sorted list are assigned to a new collector.

3) Handover Target Collector Selection: The final step in the load balancing algorithm is to select the target collector for each sensor node that must be assigned to a new collector.

Algorithm 1 Load Balancing Algorithm

```
1:  procedure LOADBALANCING(P(r), S(r), ho(r))
2:      if (|P(r)| > 1) ∧ (|S(r)| > 1) ∧ (|ho(r)| == 0) then
3:          remainder = (|S(r)|)%|P(r)|
4:          quotient = floor(|S(r)|/|P(r)|)
5:          temp = P(r)
6:          while (|P(r)|) do
7:              p_k ∈ P(r)
8:              p_k.capacity = quotient
9:              P(r) = P(r) − {p_k}
10:         end while
11:         P(r) = temp
12:         if (remainder > 0) then
13:             sorted = SORT(P(r), channelIndex, ascending)
14:             first = getFirst(sorted)
15:             first.capacity = first.capacity + 1
16:             remainder = remainder − 1
17:             while (remainder) do
18:                 next = getFirst(sorted)
19:                 next.capacity = next.capacity + 1
20:                 remainder = remainder − 1
21:             end while
22:         end if
23:         GETSENSORSFORHANDOVER(P(r))
24:     else
25:         return
26:     end if
27: end procedure
```

Algorithm 2 Sensor Node Selection for Handover

```
1:  procedure GETSENSORSFORHANDOVER(P(r))
2:      for k = 1 : |P(r)| do
3:          p_k ∈ P(r)
4:          if (|ch(p_k)| > C(p_k)) then
5:              selected = |ch(p_k)| − C(p_k)
6:              sorted = SORT(ch(p_k), RSSI, descending)
7:              while (selected) do
8:                  node = getFirst(sorted)
9:                  sorted = sorted − {node}
10:                 target = GETTARGETCOLLECTOR(node)
11:                 if (target ≠ null) then
12:                     ho(r) = ho(r) + {node.target}
13:                 end if
14:                 selected = selected − 1
15:             end while
16:         end if
17:     end for
18: end procedure
```

Subset $P(r) - p_i$ contains the possibles new (target) collectors. Within this subset, a collector is eligible to become a target collector when $ch(p_i) < C(p_i$ and it is currently allowing sensor nodes to join its network ($p_i.status = permitJoiningOn$)

as in (1).

$$\exists\ p_j \in \{P(r) - \{p_i\}\} \mid (ch(p_j) < C(p_j) \wedge p_j.status = permitJoiningOn) \quad (1)$$

Finally, the $s_i \in ho(r)$ are instructed, one at a time, to perform handover to their assigned target collector following the procedure in section III. If unsuccessful, the sensor node is removed from set $ho(r)$ on timeout of the handover timer.

IV. The Testbed

The testbed incorporates one NC with up to 4 collectors and 50 sensor nodes as illustrated in Fig. 3. The array of 50 sensor nodes is built using a combination of CC1310 and CC1350 TI LaunchPads[4] spatially distributed to achieve a balanced count of these devices in every network size that is evaluated. The sensor nodes are powered up by columns of 10 devices to facilitate the wiring and using a single 3.3 VDC power supply. A single digital multimeter[5] is used to measure the current delivered by the DC power supply to the array of sensor nodes. To accurately measure the consumption of the Sub-1 GHz radios, all jumpers on the TI LaunchPad are removed and the NC and collectors make use of a separate power source. The sensor nodes are configured to transmit frames containing their MAC and application layer statistics as payload. The data is then collected from the NC over an SSH connection for further processing and analysis.

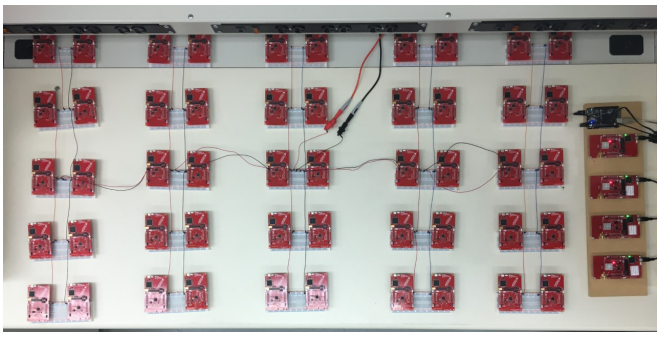

Fig. 3. Testbed with 50 sensor nodes, one NC, and 4 collectors (right side) in the UTD Laboratory to evaluate the multicollector system.

V. Experimental Results

The performance of the WSNC system is evaluated under race conditions to emulate the presence of a large number of sensor nodes using the testbed in section IV. Such conditions are generated by setting the reporting interval time of the sensor nodes to be a small fixed value[6] of 1 second across all the experiments. Different network sizes of 10, 20, 30, 40, and 50 nodes are evaluated. As an example, $N_S = 50$ is approximately equivalent (load-wise) to having 5,000 sensor nodes periodically reporting data every 100 seconds. With the chosen report time, race condition is minimal or not present when $N_S < 20$ and becomes relevant when $30 \leq N_S \leq 50$ in the single collector scenario.

The evaluation is carried out varying the network size ($10 \leq N_S \leq 50$), the number of collectors ($1 \leq noc \leq 4$), and the channel separation (chg) to account for the inter-channel interference resulting from the use of multiple collectors. In particular, two scenarios are compared: minimum channel separation (contiguous) $chg = 1$ – worst case, and maximum channel separation $chg = 127, 64$, and 43, for $noc = 2, 3$, and 4, respectively – best case.

Each experiment has a duration of approximately 34 minutes which provides a reasonable time interval to compute steady state statistics. Performance indicators are the Packet Delivery Ratio – PDR and Energy Per Packet Delivered – EPPD. Due to the unreliable nature of the wireless medium and possible uncontrolled external factors, some outliers may be present in the results.

TABLE II
Experiment Parameters

Parameter	Value
Operating mode	NBM
Modulation type	2-GFSK
Data rate	50 kbps
Frequency band	915MHz
Transmit power	0 dBm
Payload size	80 bytes
Reporting interval time	1 s
Polling interval time	6 s
Experiment duration	34 min

A. Packet Delivery Rate – PDR

The PDR is an indicator of how efficient the network is while deliverying the sensor nodes' packets to the NC. It is computed as the ratio between the total messages successfully delivered over the total message transmissions attempted by the sensor nodes. As depicted in Fig. 4, the multicollector system reduces the channel congestion and increases PDR by up to 40% ($noc = 4$) in dense and highly loaded networks relative to the single collector scenario ($noc = 1$). As expected, race conditions prevent sensor nodes from achieving high PDR due to channel congestion and packet loss. Maximum channel separation outperforms the case when $chg = 1$ and yields increasing PDR gains that are more appreciable as the network grows ($N_S > 20$). In smaller networks ($N_S \leq 10$) the PDR approaches 100% and channel separation chg has minimum effect.

B. Energy Per Packet Delivered – EPPD

Adverse channel conditions can degrade the communication quality between sensor nodes and collectors and in turn increase both the retransmission attempts and packet delivery failures. The Energy Per Packet Delivered – EPPD indicator

[4]Both devices have the same Sub-1 GHz radio, however, the PCB antenna of the CC1310 LaunchPad offers a higher gain. The CC1350 LaunchPad has also a BLE radio not used in our experiments.

[5]Not having access to a DC Power Analyzer the results are limited by the resolution of this instrument.

[6]We kept this value fixed since varying the reporting interval time is equivalent to modifying the network size.

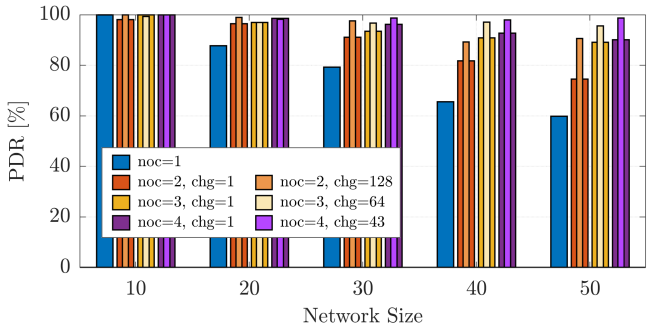

Fig. 4. Packet delivery rate as a function of network size, number of collectors (noc), and channel separation (chg).

accounts for all these radio activities and helps quantify the effort incurred (energy used) by a sensor node to deliver a single message to its parent collector. More specifically, the EPPD is calculated in two steps. First, the power signal is obtained by multiplying the current signal output from the meter with the output voltage (3.3V) of the DC power supply. Second, the total energy consumed is obtained by integrating the power signal over the observed time interval, and the EPPD is calculated by dividing the total energy by the number of messages successfully delivered to the NC.

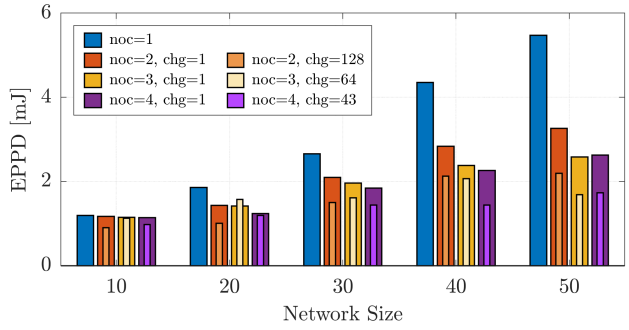

Fig. 5. Energy per packet delivered as a function of network size, number of collectors (noc) and channel separation (chg).

From Fig. 5 it can be seen that in all scenarios the EPPD tends to grow as the network size increases and the resulting increased network congestion causes more transmission attempts to be carried out by the sensor nodes. In contrast, in smaller networks, race conditions are mostly avoided and the EPPD is comparable in all experiments, regardless of the number of collectors used. Lower EPPD is achieved in most cases when maximizing channel separation compared to the $chg = 1$ case. This is the case even when the network size is small, since the energy consumption at higher frequencies is slightly less compared to that of lower frequency channels.

VI. Conclusion

The proposed WSNC system constitutes a cost-effective solution to aggregate multiple collectors in order to increase the network capacity and improve the performance of IEEE802.15.4g-based WSNs. By leveraging low cost WMCUs to realize multiple collectors connected to a single network concentrator (NC), the WSNC system offers multiple parallel radio channels in the Sub-1 GHz spectrum that can be efficiently used to reduce channel congestion and energy consuption at the sensor nodes. Maximum channel separation of the frequencies assigned to the collectors is shown to improve performance by reducing inter-channel interference. The WSNC system can be of practical use in scenarios with high counts of sensor nodes and where quality of service, energy consumption, and network responsiveness requirements are of the essence. In addition, this aggregation scheme minimizes the number of cellular gateways that are required to send collected sensor data to a remote cloud server, thus potentially reducing the cost of the IoT network infrastructure.

References

[1] IEEE Std. 802.15.4, "Wireless MAC and PHY Specifications for Low-Rate WPANs," Sept. 2011.

[2] IEEE Std. 802.15.4g, "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) – Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks," April 2012.

[3] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, Emden, Germany, July 2017.

[4] C.S. Sum, F. Kojima, H. Harada, "Coexistence of Homogeneous and Heterogeneous Systems for IEEE 802.15.4g Smart Utility Networks", *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN),* 2011.

[5] C. Orfanidis, L. M. Feeney, M. Jacobsson, and P. Gunningberg "Investigating interference between LoRa and IEEE 802.15.4g networks," *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* 2017.

[6] C. Sum, H. Harada, F. Kojima, and L. Lu, "An Interference Management Protocol for Multiple Physical Layers in IEEE 802.15.4g Smart Utility Networks", *IEEE Communications Magazine*, April 2013.

[7] F. V. den Abeele, J. Haxhibeqiri, I. Moerman and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *IEEE Internet of Things J.*, vol. 4, no. 6, 2017.

[8] A. Lavric, A. I. Petrariu and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale high-density conditions," *IEEE Access*, vol. 7, pp. 35816-35825, 2019.

[9] J. Banik, R. Arjona, M. Tacca, M. Razo, A. Fumagalli, K. Vijayasankar, and A. Kandhalu, "Improving performance in industrial Internet of Things using multi-radio nodes and multiple gateways," *2017 International Conference on Computing, Networking and Communications (ICNC)*, January, 2017.

[10] A. I. Petrariu, A. Lavric, and E. Coca, "LoRaWAN Gateway: Design, Implementation and Testing in Real Environment," *2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2019.

[11] F. V. den Abeele, J. Haxhibeqiri, I. Moerman and J. Hoebeke, "Development of Multichannel LoRaWAN Gateway for Educational Applications in Low-Power Wireless Communications," *Proc. XXVIII International Scientific Conference Electronics – ET2019*, September, 2019.

[12] K. Mikhaylov, A. Pouttu, "On Spatial Diversity for LoRaWAN: Experimental Evaluation of Performance of a Dual-Gateway Network With and Without Downlink," *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, October, 2019.

[13] R. Arjona, A. Fumagalli, C. Lee, and K. Vijayasankar, "An Experimental End-to-End Delay Study of a Sub-1GHz Wireless Sensor Network with LTE Backhaul," *2018 IEEE Global Communications Conference (GLOBECOM)*, December, 2018.

[14] Available: https://beagleboard.org/black-wireless.

[15] Texas Instruments. CC1310 SimpleLink Ultra-Low-Power Sub-1 GHz Wireless MCU, 2018. Available: https://www.ti.com/product/CC1310.

[16] Texas Instruments. SimpleLink TI 15.4-Stack Developer's Guide, 2016.