CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	A network architecture for high volume data collection in agricultural applications
Authors	Zorbas, Dimitrios;O'Flynn, Brendan
Publication date	2019-05-01
Original Citation	Zorbas, D. and O'Flynn, B. (2019) 'A Network Architecture for High Volume Data Collection in Agricultural Applications', 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 29-31 May, pp. 578-583. doi: 10.1109/DCOSS.2019.00107
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/abstract/document/8804484 - 10.1109/DCOSS.2019.00107
Rights	© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-03 15:14:07
Item downloaded from	https://hdl.handle.net/10468/8804

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

A Network Architecture for High Volume Data Collection in Agricultural Applications

Dimitrios Zorbas and Brendan O'Flynn
*Tyndall National Institute, University College Cork, Ireland
Email: dimitrios.zormpas@tyndall.ie

***Abstract*—An important requirement for Internet of Things applications is the ability to provide fast and energy efficient data collection from wireless sensors. When sensor nodes are located far from the data collection point, currently available long range protocols present challenges associated with a very low data rate and often unreliable connections resulting in excessive energy consumption related to data transmission. To address this problem, we propose a simple and energy-efficient data collection architecture for smart agricultural purposes which require wireless sensing. The architecture involves data collection from nodes located in remote fields or on animals leveraging off the use of drones as a data collection mechanism. In particular, drones can fly over the desired areas (points) and collect high volumes of data that would be otherwise difficult to transfer directly to the sink in a reasonable amount of time and using reasonable amounts of energy. We describe the different components and stages that constitute the proposed architecture emphasizing the networking component. We propose the use of different communication technologies, such as LoRa and WiFi, depending on the data collection requirements. We present an in-lab development of this architecture as a proof-of-concept as well as preliminary results for the architecture. The results reveal that the proposed solution is potentially capable of achieving data collection at high volume, however, the performance does not consider the highest spreading factors of LoRa.**

I. Introduction & Background

The concept of the Internet of Things (IoT) has been recently expanded to many application domains. One of these domains is the area of precision agriculture, where nodes are placed in smart farming environments to capture specific behaviours related to the quality of crops and the well-being of animals [1]. In this type of network, depending on the application, the nodes wake up every few seconds or minutes in order to take a measurement (e.g., temperature) and then transmit the data to a near base station. This is a typical network architecture which may work well, for instance, in a smart city scenario [2] where there is seamless connectivity and unlimited power supply.

The main issue of deploying such a network in an agricultural environment is the lack of a robust, constant and reliable connection between the nodes and the base station. These agri-networks are usually deployed in rural areas where even the cellular coverage is poor. In previous work [3], [4], we proposed the use of drones together with the LoRa communication protocol as an efficient mean of collecting the data. Indeed, since monitoring in agricultural applications is not time-critical (daily transmissions are considered sufficient), the nodes can store their data in their memory and transmit them once the network becomes operational. Drones can assist in this data collection process as mobile gateways (sinks). We observed that since most of the drone flying time is spent travelling from one node to another, only a long range protocol like LoRa can be suitable to cover large geographical areas. This solution works well – in terms of node energy conservation and coverage scale – when a few only kilobytes of data need to be transmitted per node (e.g., a temperature measurement every ten minutes over a single day). However, the data transmission may take a considerable amount of time (many hours) when the data size increases in the scale of megabytes (e.g., data coming from accelerometers on animals). This significant data collection time makes the solution impractical considering the current maximum flying time and range of the drones, which are based on battery size.

In order to solve the aforementioned issue, we propose a solution which can be applied to applications that require a transfer of large amounts of data. The solution employs a multi-radio communication architecture both for the nodes and the drones, as well as for the drone command center. In particular, we still use the LoRa protocol as a node-to-command center communication solution, however, we restrict its usage to transmit requests and geographical positions only and not data. The data is transferred using higher data rate protocols, such as Bluetooth or WiFi, once a drone comes closer to the node. We present the requirements of using this solution in an agricultural scenario explaining the advantages and the drawbacks. Apart from this, we develop a prototype to evaluate parts of the proposed architecture and we present preliminary results.

The present work is one of the first works that utilize multiple-radio devices to achieve data collection using drones. Even though data collection using autonomous vehicles as data aggregators is not a new concept [5], [6], most of the works described in the State of the Art deal with the problem of finding optimal vehicle trajectories [7], [8] or of minimizing the drone's lifetime [9], [10]. Specific networking solutions or practical considerations that can considerably decline the network performance are usually ignored. The closest work to ours considers an Unmanned Aerial Vehicle (UAV) that can fly over vessels (boats) in order to collect data [11]. However, the authors use the LoRa protocol as a communication mean between the vessels and the UAV which, as they mention, can degrade the performance.

II. Smart farming network architecture

A. Overview

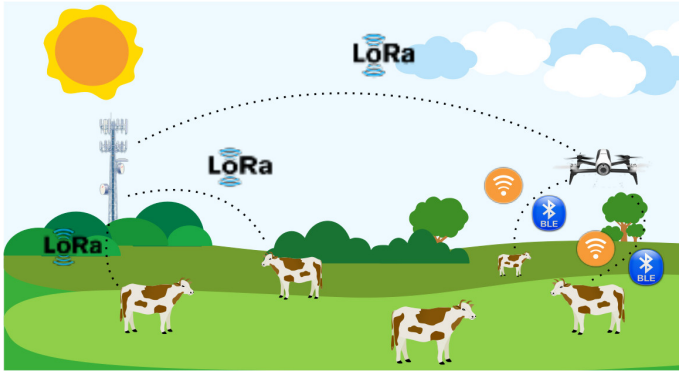


Fig. 1. An overview of the examined network architecture.

In this section, we present an architecture for high volume agricultural data collection focusing on the networking part and the synergies between the system components.

An example of the proposed architecture is displayed in Fig. 1. It consists of three entities; (a) the end node devices which are placed on animals or in fields, (b) the drones, and (c) the command center (CC). The nodes are energy constrained devices with sensing, storage, and communication capabilities. They generate data, send requests to the CC, and transmit their data to the drones. For this purpose, they are equipped with multiple communication modules; one for long range transmissions and one for shorter but of higher data rate transmissions. If the nodes are mobile, they must also have a positioning system to monitor their location. The drones are used to travel over the end devices and collect the data. They can have similar communication capabilities to communicate with the CC and the nodes. The CC is responsible for gathering the requests, replying to the nodes, scheduling the requests, and assigning activities to the drones.

The sensor nodes wake up, take a measurement, and store the data in their memory. This activity may be periodic or triggered by an event depending on the particular use case. The data size as well as the frequency of the measurements depends on the application. For instance, field monitoring generates much less data compared to animal monitoring. This is because a typical field node measures parameters such as soil temperature, moisture, light intensity, air temperature and humidity, which all together can be stored in a 50-byte packet. Assuming that these measurements are taken every 10 minutes, the accumulated amount of data in a week is less than 50KB. This amount of data can be easily sent to the CC (in real time or in bulk) over a low data rate protocol, like LoRa or NB-IoT. However, switching to an animal monitoring application, we can observe that the amount of data is much higher. Since an animal (e.g., a cow) is moving most of the time (even partial movement of a body part) accelerometer data is continuously generated. This implies data generation of some hundreds of KBytes within a time window of a single day or a few Mbytes within a week. In LoRa, in the best case scenario without taking into account duty cycle limitations and re-transmissions due to collisions, 1MB of data can be transferred in about 2 hours. The data transfer can be made faster using the NB-IoT protocol, but in this case, we have to rely on third-party providers and consider additional expenses.

B. Node to Command Center communication

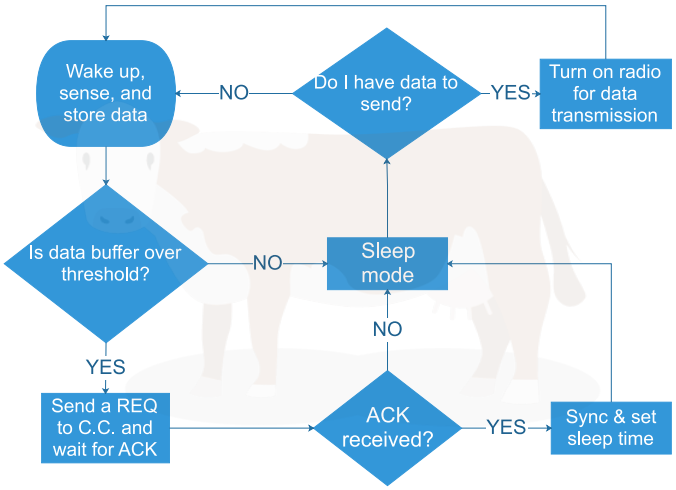


Fig. 2. Node request generation flowchart.

To tackle the issue of the high data volume transfer over a low data rate protocol, we propose a three-phase data delivery approach. The flowchart of the first phase is illustrated in Fig. 2. Every node in the system can send a request for data collection to the CC once its data buffer is above a predefined threshold. This request is sent over a long range protocol and contains information about the node id (2 bytes), the type of the message (1 byte), and optionally the location of the node (8 bytes). Extra information such as the data capture times, the packet length as well as security credentials can be also included[1]. The CC must receive the request, find an available drone to assign the job, and send a reply (ACK) to the node with the estimated service time and optionally some synchronisation information. The CC's response must be as fast as possible since during this period of time the node has its radio on and, thus, consumes more energy. The node can go back to the sleep mode once it receives the CC's response. It will wake up at the indicated time to send the data to the drone using a shorter but faster communication protocol. We will describe this process in the next subsection.

The node-CC communication must be as reliable as possible. Both the node's request and the CC's response must be delivered properly, so that there is no waste of energy or time. However, transmissions in the default LoRa MAC layer are ALOHA-based. This practically means that there is no guarantee that a transmitted packet will be finally delivered. The use of an acknowledgment on the CC side ensures that the

[1]We must note here that if LoRaWAN is used, a security mechanism is already provided.

request has been received but what happens in the case where the acknowledgment is lost? In this case, the node will keep sending requests wasting its energy or will go to sleep mode without, however, knowing that a drone is coming to collect the data. To avoid this kind of situation, we propose the use of a re-transmission policy whenever the node does not receive an acknowledgment from the CC. The maximum number of re-transmissions can be determined by the network density and other environmental parameters (e.g., path-loss, external interference etc.). On the other hand, the CC can re-send an acknowledgment multiple times to increase the probability of acceptance. We must mention here that the CC does not know if an ACK has been delivered because acknowledgements are not acknowledged. For this reason, a burst of ACKs can be sent one after the other.

Nevertheless, a problem that can degrade the performance of this re-transmission policy is the duty cycle restriction [12]. In EU a node has to wait 99% of the time before starting to transmit again. Since LoRa trades distance with data rate, higher spreading factors (SF) present extremely high transmission times (due to the low data rate). For example a node which is far away from the gateway and uses SF12[2] has to wait about 28.58 seconds in order to re-transmit a packet of 15 Bytes (assuming a channel bandwidth of 500KHz). It is understood that this is a huge waste of time and can lead to desynchronizations. Hence, high spreading factor values are less practical in our case unless a *Listen Before Talk* method is used [13] where the duty cycle limitation is not applied.

Apart from the duty cycle limitation, higher spreading factor values are not practical from another perspective. These high SF values are mostly used by nodes that are very far from the gateway. This distance may be up to some kilometers long in rural areas [14]. Considering the current battery lifetime of most commercial drones, it is not possible to approach areas located further than a few hundred meters away from the base [3]. On the contrary, lower SF values (SF7-9) can easily provide coverage to considerably high and, at the same time, drone approachable areas, without remarkably compromising the waiting time before re-transmissions.

C. Node to drone data transfer

A node can send its data to a drone once the latter has arrived at a location close to the former. To achieve this, a number of conditions must be met. First of all, the two devices must be synchronized according to the same global clock, so that the node wakes up approximately at the same time when the drone arrives at the indicated position (preferably within a difference of a few seconds). The time tolerance depends on the application and the remaining energy of the node, but it is obvious that the shorter the time difference, the better the performance. This clock is initiated by the CC and the synchronization is maintained during the node-CC communication. The second condition is related to the location of the drone. This location must permit a reliable ad-hoc connection between a drone and a node throughout the data delivery process. However, the drones must fly high enough to avoid obstacles, not scare the animals and, at the same time, comply with the regional aviation regulations. The actual position of the drone is decided by the CC based on empirical data as well as on previous data collection attempts. However, minor position corrections can be made by the drone during the process.

[2]LoRa accepts spreading factor values from 7 to 12.

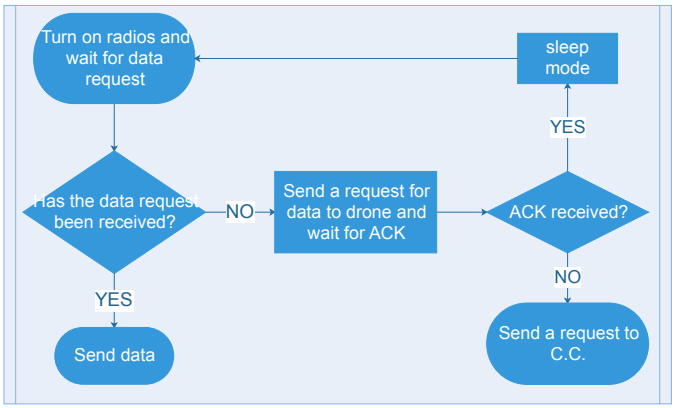


Fig. 3. Node to drone request flowchart.

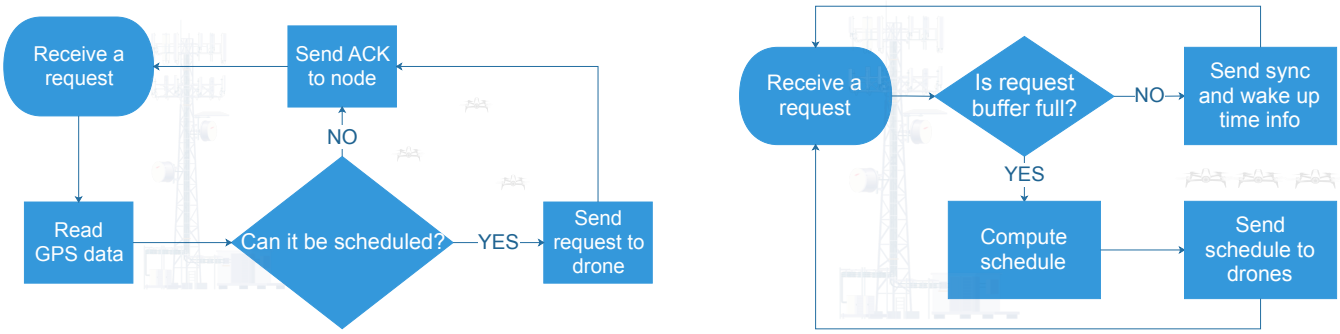
Due to eventual slight desynchronizations of the nodes or delays of the drone arrival time, a node may wake up too early to transmit the data. For this reason, a node-to-drone means of communication is needed to re-calculate the data delivery time. This sub-process is part of the node request process and is described by the flowchart of Fig. 3. According to this sub-process, a node can wait for a maximum time before sending a request for data collection to the corresponding drone. We note that the node is aware of the drone id since it has received this information in the CC's ACK message. The node must include its location information in this request, so that the drone can approach it in the case the node has been considerably moved from the initially indicated location. The drone must reply with an ACK message containing the new arrival time. If no reply is received within a time limit, the node must restart the process, initiating a new request to the CC.

In the reverse scenario where the drone arrives on time at the indicated location and the node has not woken up yet, it has to wait until it receives a new data request from the node. If it does not happen within a time limit, the drone can proceed to the next node or return to the base.

D. Command center operation

The command center operation is critical for the system reliability and efficiency. In this paper, we distinguish two scheduling architectures, named *Online* and *Offline* respectively, that they differ in the way they handle the node requests. The operation of the two processes is presented in Fig. 4.

In the Online version, the requests are handled one after the other in a first come – first serve order. Every time a request arrives, a scheduler is responsible to check whether an active drone can provide the service by calculating its remaining



(a) *Online* request handling process.

(b) *Offline* request handling process.

Fig. 4. Command Center *online* and *offline* request handling flowcharts.

flying time. If not, the job is assigned to a new available drone. The CC must send an ACK message to the node containing, the message type (2 bytes), the estimated arrival time (so that the node sleep time can be computed) (4 bytes), the drone id (2 bytes), and optionally clock synchronization data (4 bytes for clock correction [15]).

The Offline version handles the requests in batches and assigns the corresponding jobs to drones once the buffer is full or after a predefined amount of time. A practical problem in this case is that the CC cannot include any specific information about the drone arrival time nor its id in the ACK message. This must be done at a predefined future timing when the complete job schedule is available. We must note the job scheduling problem is not examined in this paper. This type of problems can be treated as operation research problems and a number of solutions exists in the literature [16], [17].

Each of these two approaches has its own advantages and disadvantages in terms of reliability/practicality and energy/budget efficiency. The Online approach is more distributed and can better handle drone or node failures. Moreover, the nodes remain active only for a short amount of time until they receive the ACK message. However, the drone scheduling is not optimal and, thus, a higher number of drones (budget) is required to serve multiple requests at different geographical areas. The Offline approach covers the weaknesses of the Online version, but it requires a second round of communication between the nodes and the CC to advertise the service times. Apart from this, the data delivery comes with some extra delay due to the collection of the requests and the computation of the schedule.

III. Evaluation & Discussion of the Results

In this paper, we focus on the evaluation of the networking part of the proposed architecture. In particular we examine the reliability and the energy efficiency (in terms of active time) of the node-CC and node-drone communications.

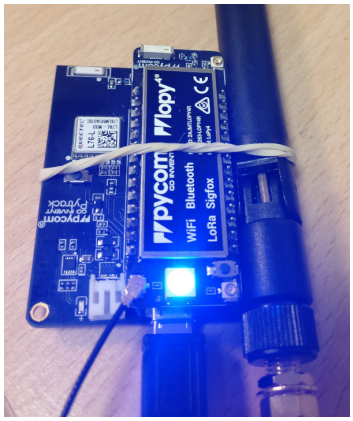
A. Setup

Due to the lack of a proper large space, the experiments were conducted indoor, in a lab room of $25m^2$ using an end node, a drone, and a device used as the CC (see Fig. 5). All the three devices were equipped with LoRa, WiFi, and Bluetooth (BLE) modules[3]. Due to the limited space, we were not able to assess all the drone parameters, like different altitudes and multiple data collection points. However, through these experiments we were able to assess the impact of a potential external interference to the node-CC communication and, moreover, to measure the overall time needed to perform a reliable data exchange between the two parts. Finally, we recorded the time needed to send the data to drone once it reaches the desired destination. These two time periods, the node-CC and the node-drone communication, are important for the node lifetime and, thus, the longevity of the application. Each experiment was executed 20 times and the average results are presented along with the minimum and maximum values. All the experiment parameters are summarized in Table I.

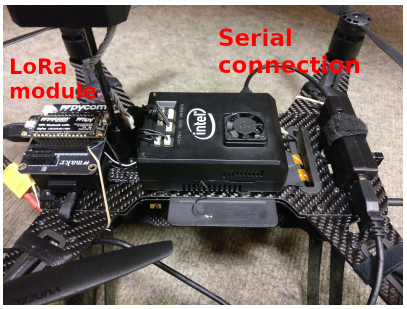
B. Evaluation of the node-CC communication

In the first set of experiments we measured the total required active time of the node to communicate with the CC as well as the number of re-transmissions. The active time corresponds to the time needed to prepare a packet, send a request to the CC, receive an acknowledgment, and go to sleep mode. We evaluated the proposed method of multiple retries in the case of a packet loss using different LoRa spreading factors, with or without the presence of external interference. In order to generate interference for the purposes of the experiment, we placed a 10-node LoRa network along with a separate gateway co-located at the experiment site. All the nodes used the same LoRa settings (SF, bandwidth, and transmission power) with the main network. Just for the needs of the experiments and in order to generate a considerable amount of interfering traffic, we exceeded the maximum allowed duty cycle allowing the external network nodes to transmit packets with a maximum duty cycle of 3.25%. The experimental results are depicted in Table II and they show very low active times when no interference is present. Indeed, even with SF12 the total active time is little higher than a half of a second. On the other

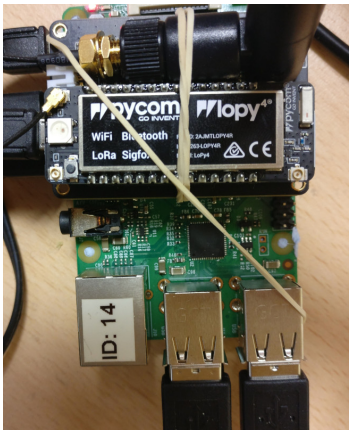
[3]Due to the current restricted BLE functionality of the tested devices, we conducted the experiments using WiFi only.



(a) Pytrack accelerometer device with Pycom LoRa and WiFi modules.



(b) Intel Aero Drone (WiFi ready) with an extra Pycom LoRa module.



(c) A LoRa gateway connected to a Raspberry Pi serving as the Command Center.

Fig. 5. The experiment's devices.

TABLE I
EXPERIMENT'S PARAMETERS

Parameter	Value
LoRa	
Bandwidth (BW) - Coding Rate	500 KHz - 4/5
Spreading Factors	7-9, 12
Region	EU868
Transmission power	2 dBm
Preamble symbols	8
Request packet size	15 Bytes
CC ACK packet size	8 Bytes
Max node re-transmissions	9
CC re-transmissions	2
Node and CC duty cycle	1%
WiFi	
Mode & Security	IEEE802.11n, WPA2
Buffer size	2048 Bytes
Sensing data volume	1.2, 3.5, 5.2 MB
Drone-node distance	∼4m
Node antenna	Built-in
Power consumption	137mA [18]
External LoRa network	
Number of nodes	10 + 1
Node duty cycle	3.25%
Packet size	50 Bytes
Re-transmissions	No
Acknowledgements	No

hand, the re-transmission mechanism worked well in presence of interference, however, the active times are much higher. In the case of SF12, the maximum recorded time was almost 30 seconds with only one re-transmission. It is straightforward that the solution is not scalable in this case. One of the positives of this experiment was that due to the small size of the acknowledgments, we recorded only 2 ACK packets lost.

C. Evaluation of the node-drone communication

In the second part of the evaluation, we measured the active time of the node in order to send the stored data to the drone. This time includes the time needed to connect to the drone's WiFi network, establish a connection, transmit the data, and close the connection. The results are depicted in Fig. 6 and reveal that there is a required amount of time to connect to the network and establish a connection which is about 4 seconds. However, during this interval the node remains mostly in idle mode consuming considerably less energy. Assuming that the termination of the connection is negligible (a few museconds), the rest of the time is dedicated to the data transmission. Apparently, the larger the data size, the longer the transmission. However, we captured a slightly lower data rates for the large size file (2.7Mbps) compared to the small size file (3.2Mbps). The captured data rate was much lower than the maximum supported rate (150Mbps) which is mainly due to the absence of an external WiFi antenna and the co-existence of other WiFi networks. The approximated average energy consumption for the data transmission was 1.55, 5.26, and 8.03 Joules, for the small, medium, and large data size, respectively[4].

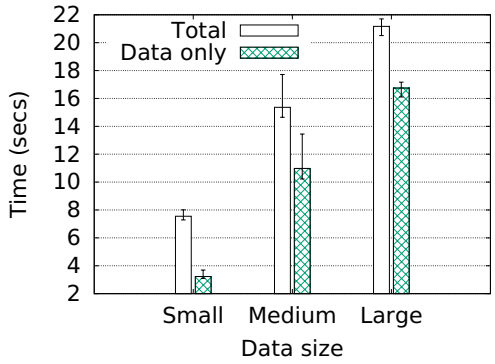


Fig. 6. Node active time to transmit data (Small = 1.2MB, Medium = 3.5MB, Large = 5.3MB).

We must note that the packet size plays an important role in the data delivery time considering such a long radio link. Transmitting large packets helps to reduce header

[4]using the manufacturer's datasheet values

TABLE II
TOTAL ACTIVE TIME (T) IN ms AND NUMBER OF RE-TRANSMISSIONS (R)

SF	Without interference						With interference					
	Avg T	Min T	Max T	Avg R	Min R	Max R	Avg T	Min T	Max T	Avg R	Min R	Max R
7	63.5	61	72	0	0	0	653.5	61	2405	0.3	0	2
8	82.4	80	84	0	0	0	1727.2	81	7987	0.6	0	3
9	117.2	116	119	0	0	0	2699.7	115	7356	0.5	0	1
12	555.3	550	567	0	0	0	8383.6	552	29456	0.35	0	1

overhead, but may have adverse effect on loss rate due to corruptions in the radio link [19]. Apparently, this affects the energy consumption as well, thus, a trade-off between energy consumption and data delivery time exists. However, this is an application-based trade-off which may depend on the current link characteristics, such as distance, environmental conditions, and external interference.

IV. CONCLUSIONS & FUTURE WORK

In this paper, we proposed a network architecture for the collection of large volumes of data, a problem that typically appears in an agricultural environment. Due to the low data rate capabilities of the current long range protocols, we proposed the use of drones as routers to transfer the data from the nodes to the base. To do so, we considered a dual-radio communication consisting of the LoRa protocol for low-cost and low-size node requests and the WiFi or Bluetooth for the shorter communication between a node and a drone. We gave details about how the data collection can be realized, how reliability can be achieved, and we mentioned the eventual drawbacks of such a solution. Experimental evaluation results of the networking part of the architecture showed that even in the presence of high external interference the proposed re-transmission mechanism works satisfactorily, however, it is hard to be applied using high LoRa spreading factor values.

In the future, we plan to extend the experimental evaluation considering additional system parameters, like multiple nodes, realistic drone heights, and to assess the node-drone LoRa communication. We also plan to use a Listen Before Talk method for higher spreading factor values and compare it to the native LoRa protocol. Furthermore, part of the work is to measure the actual total energy consumption of the node for different scenarios and packet losses. Finally, we intend to deploy the proposed architecture in a real farming scenario.

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

It has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567.

REFERENCES

[1] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[3] D. Zorbas and B. O'Flynn, "Collision-free sensor data collection using lorawan and drones," in *Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, Oct 2018, pp. 1–5.

[4] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, B. O'Flynn, and D. Pesch, "Free - fine-grained scheduling for reliable and energy efficient data collection in lorawan," 2018, arXiv:1812.05744.

[5] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sen. Netw.*, vol. 8, no. 1, pp. 7:1–7:31, Aug. 2011.

[6] N. R. Zema, N. Mitton, and G. Ruggeri, "A gps-less on-demand mobile sink-assisted data collection in wireless sensor networks," in *2014 IFIP Wireless Days (WD)*. IEEE, 2014, pp. 1–3.

[7] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, June 2018.

[8] C. Wang, F. Ma, J. Yan, D. De, and S. K. Das, "Efficient aerial data collection with uav in large-scale wireless sensor networks," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 2:2–2:2, Jan. 2016.

[9] R. I. da Silva and M. A. Nascimento, "On best drone tour plans for data collection in wireless sensor network," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: ACM, 2016, pp. 703–708.

[10] M. Ben Ghorbel, D. Rodriguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Energy efficient data collection for wireless sensors using drones," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, June 2018, pp. 1–5.

[11] C. A. Trasvina-Moreno, R. Blasco, A. Marco, R. Casas, and A. Trasvina-Castro, "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring," *Sensors*, vol. 17, no. 3, 2017.

[12] ETSI ERM TG28, "Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with power levels ranging up to 500 mw," *European harmonized standard EN*, vol. 300, no. 220, p. v2, 2012.

[13] M. O. Farooq and D. Pesch, "A search into a suitable channel access control protocol for lora-based networks," in *IEEE 43rd Conference on Local Computer Networks (LCN)*, Oct 2018, pp. 283–286.

[14] H. Linka, M. Rademacher, O. G. Aliu, and K. Jonas, "Path loss models for low-power wide-area networks: Experimental results using lora," *VDE ITG-Fachbericht Mobilkommunikation*, 2018.

[15] FUOTA Working Group of the LoRa Alliance Technical Committee, "Lorawan application layer clock synchronization specification v1.0.0," Sep 2018.

[16] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[17] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, "Chapter 2 time constrained routing and scheduling," in *Network Routing*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1995, vol. 8, pp. 35 – 139.

[18] Pycom, "fipy datasheet v1.0," https://docs.pycom.io/.gitbook/assets/fipy-specsheet-1.pdf, 2017.

[19] J. Korhonen and Y. Wang, "Effect of packet size on loss rate and delay in wireless links," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3. IEEE, 2005, pp. 1608–1613.