Article

SpaceSheep: Satellite communications for ovine smart grazing

Pedro Gonçalves [1,*] Daniel Corujo [2]

[1] Instituto de Telecomunicações and Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro; pasg@ua.pt

[2] Instituto de Telecomunicações e Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro; dcorujo@ua.pt

* Correspondence: pasg@ua.pt; Tel.: +351 234377900

Simple Summary: Animal monitoring through electronic means requires a continuous Internet connection to guarantee timely human intervention when needed. Internet access depends on the coverage of the communications network, which does not always exist in agricultural and rural environments, as is the case on the slopes of the Douro River. This work documents the optimization process of a monitoring and alarm generation system, and its integration with a satellite communications system, to guarantee the sending of alarm messages in places without Internet access coverage. Results functionally demonstrate the viability of the solution for timely delivering alarms with an acceptable expected operating cost.

Abstract: The application of IoT-based means to support pastoralism activity allows smart optimization of livestock operations and improves the activity's efficiency. The use of autonomous animal control mechanisms frees the shepherd to carry out other tasks. However, human intervention is still needed in cases such as system failure, bad or unpredicted behavior of the animals, or even in case of danger to the welfare of the animal. This work documents the enhancement of an alarm generation system, initially developed within the scope of the SheepIT project, for monitoring animal behavior and equipment, which warns the human operator of the occurrence of undesirable events that require his intervention. In this case, special attention was given to use case scenarios in places without Internet access, such as rural areas. Therefore, the system was integrated with a satellite interface, as a way of guaranteeing the timely delivery of alarm messages. As a way of ensuring an acceptable operating cost, considering the communication costs of this type of communication, the system was further optimized in terms of message encoding. The paper assesses the overall performance of the system, evaluating its scalability, and comparing the efficiency gains from the optimization, as well as the performance of the satellite link.

Keywords: Animal monitoring, IoT, Alarm systems, Satellite communications

1. Introduction

ICT technologies' application to support livestock activities started as a strategy to increase productivity [1,2] and reduce the environmental impact of the activity. The tasks automation and the monitoring of the processes' evolution, reduces the cost of labor, allows the close monitoring of electronic devices and, consequently, considerably impacts the efficiency of livestock activities.

Within the scope of the SheepIT project [3], an autonomous system for the postural conditioning of ovines was developed, as an animal-based weeding mechanism, allowing the sheep to graze within a vineyard without threatening the vines or grapes. Sheep wear smart collars connected to an Internet of Things (IoT)-like network, composed of beacons and gateways. The collars collected the posture data, allowing actuators to apply conditioning measures, with the gathered data being reported though the communication infrastructure. The data's evolution was then analyzed, through the usage of implemented analytics tools. Even though having been developed as an automated conditioning mechanism, the system still requires human supervision and maintenance, in order to ensure animal and crop safety: systems fail and lack maintenance; animals are often attacked and abducted and animal wellbeing requires the human supervisor to monitor animal behavior in offline and to intervene.

SheepIT field-trial tests [4] demonstrated the importance of automating the shepherd's job, since labor costs were reduced by freeing the shepherd to realize other tasks in the vineyard. In addition, the use of animal monitoring tools allows identifying and predicting a very wide range of events related to animals [5–8].

In order to optimize the shepherd's pastoral activity, and allow supervision to be done remotely, an application for monitoring events associated with animals and equipment was developed, integrated with the SheepIT communications gateway [9]. The monitoring application keeps watch on collars and beacons communications [10], and sends alarms whenever human supervision is needed.

An alarm system depends on fast communications to effectively operate, to be able to trigger the necessary corrective measures. Such capability is not taken for granted in rural areas, which suffer from low coverage from radio technologies such as Wireless Local Area Networks (WLAN), and even mobile networks [11]. Currently, different technologies targeting greater coverage in IoT scenarios [12] are available, namely as NB-IoT [13], Sigfox [14], amongst others [15,16]. Nonetheless, these technologies are mainly only available in or around urban centers, with rural areas still left with extensive coverage hiatus or with only limited access to all the technologies' capabilities that would be more useful in IoT use cases [10]. This paper documents an extension done over the SheepIT system, where a satellite communications interface was added to address the lack of coverage while providing sufficient performance and availability of the alarm mechanisms.

The paper is structured as follows. Section 2 described the implemented system. Section 3 assesses the communications latency and bandwidth used, with results discussed in section 4., Finally, section 5 concludes the paper and points out future work.

2. Materials and Methods

SheepIT was originally built to provide an animal-based weeding process for vineyards, using a posture conditioning collar that prevents them to eat the vine leaves and grapes. The collar mechanism was worn by the animals and integrated in a wireless sensors network (WSN) [17], capable of monitoring the animals' behavior [18] as well as their location [19]. This WSN, which is composed by a set of fixed nodes along with a gateway that aggregates and filters obtained information, can localize the animals within its infrastructure, and to interconnect it with a cloud-hosted system. The cloud hosted system can receive the monitoring data and perform long-term analysis over it, assisting in the determination of trends, predicting behavior, and generating alarms.

While grazing within the vineyard, animals are continuously tempted to also eat the vine leaves and their fruits. The collar detects the movement of the animal biting the leaves and compares the height of the neck during this process. As soon as the biting movement is detected above the maximum height threshold, the system is activated, and starts the conditioning process. The analysis of the animal behavior [4] over time evidences the continuous attempts of the animal to violate the maximum height, and that the collar avoids animal attack to vine leaves and grapes.

2.1. Satellite link integration

The main use case from SheepIT considers animal grazing in the Douro region, at Quinta da Ervamoira, an Adriano Ramos Pinto vineyard[1], a place where the geographical relief is especially rugged [20] and where cellular communications have a very poor coverage [21], inhibiting the communications between the WSN and the cloud [11], thus compromising alarm messages dispatch. This is an ideal deployment setting for satellite-based communications [22], which allow coverage to be provided in areas that are technically challenging, or economically unfeasible, for other technologies. Nonetheless, one important characteristic of satellite-based communications is the cost per bit. Pricing models are associated with the data volume contracted by the customer, but mobile satellite communications service providers usually place data transmission costs between 4.79 and 9.58 euros per Mbit.

As a result, the evolution done to the gateway of our previous solution [9], besides integrating and assessing a satellite communications interface for sending alarm messages, was also optimized to reduce the amount of data sent, by optimizing the encoding of animal monitoring messages as well. In its initial form, a RabbitMQ message broker [12] producer API ensured the communication between the application running in the cloud and the gateway, with the information encoded in JavaScript Object Notation (JSON) [23]. To evaluate the best high-availability and fault-tolerance solution for processing messages, we tested the Apache AVRO [24], MessagePack [24,25] and Protocol Buffers [26] encoding APIs.

Our testbed setup, illustrated in Figure 1, made use of the satellite network belonging to the EchoStar Mobile operator, connected via a Hughes 4200 portable data terminal [27]. This terminal acted as a field concentrator interfacing

[1] - Quinta de Ervamoira, Vila Nova de Foz Côa - 41.02107689068905, -7.110593416910681

the edge computing device (collecting the data from the sensors in the animals' collars) with the SheepIT cloud service. The use of a satellite link allows for the SheepIT system to be deployed in rural and remote unconnected zones, thus guaranteeing 24/7 smart monitoring.

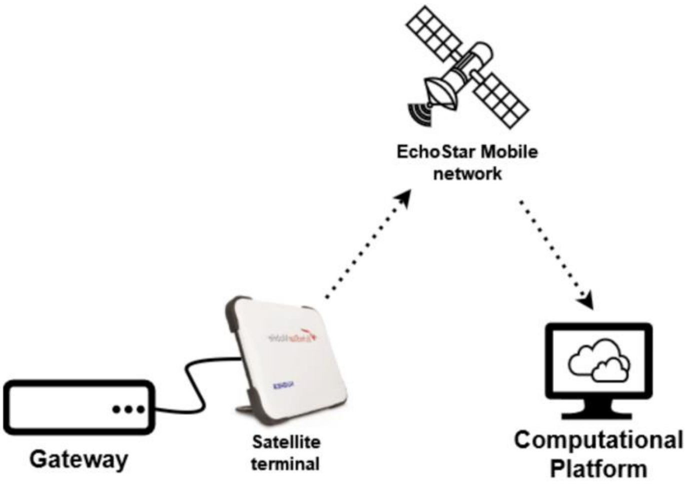

Figure 1. Satellite network integration.

2.2. *Pasture alarm system*

The alarm system was implemented in the SheepIT gateway monitoring the periodic communications forwarded by fixed WSN beacons, reporting the data sensed by the collars. Those periodic communications allow the detection of various events of interest related to animal behavior and health, as well as other events that require prompt intervention by the supervisor of the herd.

Both the gateway and the cloud system generate alarms, summarized in Table 1 along with their type, source, the context in which they are generated and size (bytes). One of the alarms generated by the system has to do with the excess of infractions carried out by an animal, that should notify the human supervisor to remove the animal form the vineyard weeding job, since there are animals that do not accept to be conditioned.

Table 1. System alarms

Type of alarm	Alarm Source	Alarm context	Message size (bytes)
Battery	Gateway	Battery level of the network nodes goes below a minimum threshold	1134
Absence	Gateway	Network node is no longer detected after several communication cycles	1147
Infraction	Gateway	Animal crossed a threshold of number of infractions per unit of time	1161
Panic	Gateway	A pattern of accelerations from multiple elements of the herd is detected in the same period	1159
Inactivity	Gateway	Detection of a pattern of collar inactivity, evidencing that the animal may have removed the collar.	1134
Health	Cloud	Prolonged decrease in an animal's activity was detected	-

SheepIT network nodes are battery powered, and they periodically report their battery status in order to allow the system to monitor their charge, and to inform the supervisor about it. The collars they are the key elements of the system, which guarantee that animals do not harm the vineyards. Moreover, as soon as the battery value drops below the minimum established threshold value, an alarm is triggered by the system to the supervisor. Besides batteries running out, two more types of such events can happen: equipment failure and inactivity. The first can be associated with the equipment having become inoperable, in which case the animal becomes free to endanger the vine leaves. The second has to do with the equipment was abandoned on the ground, leaving the animal free as well, and that is detected by a pattern of sensor-detected accelerations, which are not very common in the normal animal behavior.

The panic alarm is the last type of locally generated alarm. In particular, the gateway undergoes a continuous comparison between the collars' accelerations and the baseline acceleration values for each of the animals. This allows the system to detect disturbances affecting the herd as a whole, such as strangers or other animals (e.g., stray dogs) interacting with it.

The information sent in alarm messages, illustrated in Table 2, include the following attributes: a timestamp with the date at which the alarm was generated, the Device Type identifying the type of device that triggered the alarm; the ID that identifies the device's identifier; the Alarm Type that defines the type of the alarm situation; the Priority indicating the priority level of the alarming situation; and Additional Information, an optional field with complementary information.

Table 2. Examples of alarm notifications.

Timestamp	Device type	ID	Alarm type	Priority	Additional information
Sat Nov 20 07:22:36 2021	Beacon	1	Battery	Low	Battery: 18 (%)
Sat Nov 20 07:36:12 2021	Collar	2	Infraction	Low	#Warnings = 21 (in 3 min)
Sat Nov 20 08:19:47 2021	Beacon	1	Battery	High	Battery: 6 (%)
Sat Nov 20 08:37:14 2021	Collar	2	Equipment	High	

By its turn, the cloud application monitors the herd data and it generates alarms, for example, signaling a continuous decrease in animal activity, thus indicating a potential illness. Since those messages are generated in the cloud and are not exchanged in the satellite/WiFi links, their size is irrelevant, and therefore their encoding was not optimized.

3 - System evaluation

The monitoring system was tested and functionally validated, with its performance assessed in terms of latency and generated traffic volume. Additionally, tests were carried out with different numbers of events to determine the scalability of the system depending on the size of the herd to be monitored.

To undergo tests on the system, A Raspberry PI 3 Model B+ single board computer was used to implement the gateway, and the cloud application component was hosted in a virtual machine enabled with 8GB RAM and 2 cores, with the Ubuntu 20.04.3 LTS server Linux operating system, hosted at the datacenter of the Instituto de Telecomunicações.

The performance of the signaling encoding APIs translates especially in terms of the volume of information to transport the information and the latency of the communications.

3.1. Alarm transfer cost and latency

The latency of information transfer is a critical factor when it comes to an alarm sending system. The latency of the JSON encoded alarm system was measured and compared with WiFi connectivity latencies. After the satellite connection was established, the messages used for testing simulate the transport of the collar information received by the WSN to the cloud hosted application. Regarding the first test, two metrics were used to assess the daily volume of traffic exchanged; a) the number of connected collars: (1, 2, 5, 10, 20, 50, 100, 200, 500, 1000); and b), the reporting periodicity (s) (30, 60, 90). In this scenario, the same number of messages was simulated for each combination of the previous two metrics, and the duration of the experiment varied according to the periodicity, with higher duration for the period of 90 seconds. From the results obtained, the amount of data generated in a 24-hour interval was extrapolated and its cost was calculated.

The results show an increase in the amount of data generated with the increase of the number of collars and the decrease of the period, and the maximum volume of data is achieved for 1000 collars and a period of 30 seconds, with a daily volume of approximately of 1.89 GB.

In terms of financial cost, considering the data transmission cost of this network (of 4.79 to 9.58 € per Mbit), even the unrealistic scenario of a herd with a single sheep would have a significant cost: for a period of 90 seconds, the daily cost would be between 9.0 × 8 × 4.79 ≈ 345 € and 9.0 × 8 × 9.58 ≈ 690 €.

Regarding the second test, the latency of the system for the transmission of the collar information was registered at different times of the day, both using WiFi connectivity and the EchoStar's satellite link. This experiment was repeated 10 times for both 1 and 100 collars. Moreover, the results were registered with a 95% confidence interval, using a t-distribution.

As for the results test, Figure 2 shows the variation in the time taken to send information. As can be seen, in both cases the latency increases with the message size. Both technologies show some fluctuation throughout the day, although this would have no impact when applied in a realistic scenario.

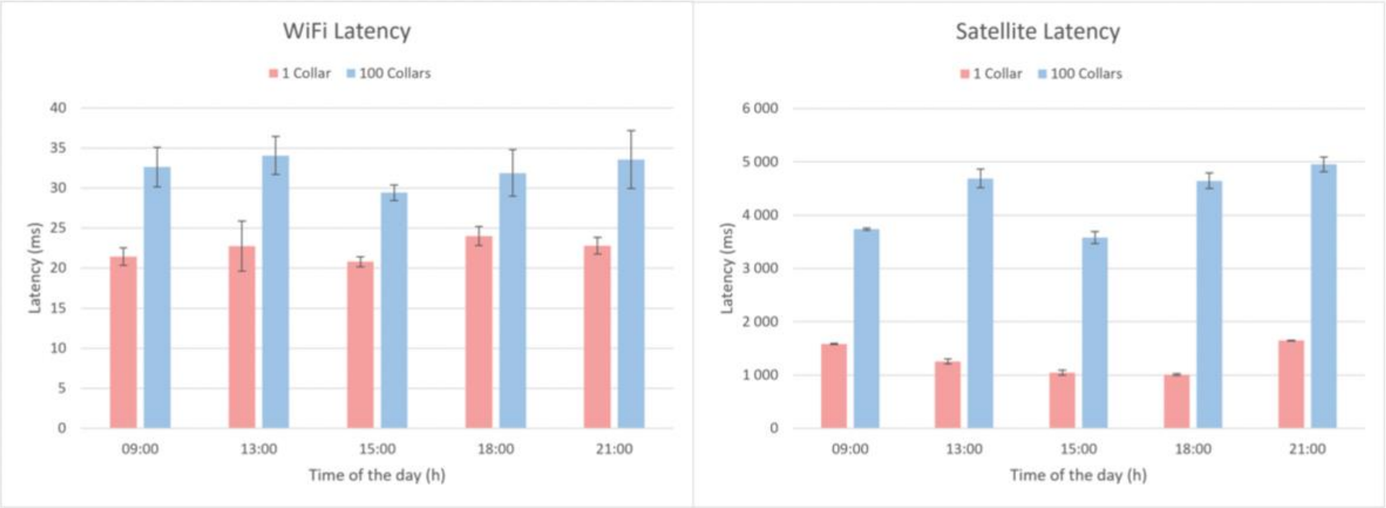

Figure 2. System alarm dispatch latency at several hours of WiFi network (left) and a satellite network (right).

3.2. Alarm encoding evaluation

The alarm encoding evaluation focused on the assessment of three serialization formats, namely Apache Avro, MessagePack and Protocol Buffers. To do so, for each existing type of alarm, a varied number of messages was generated through an alarm simulator created in the gateway. The maximum number of messages simulated for the two types of devices (collars and beacons) is different, since a real system uses a reduced number of beacons to cover the pasture area while monitoring a herd that is typically composed of a greater number of animals.

The tests were repeated 10 times for each combination and the average was registered with a 95% confidence interval, using a t-distribution. The performance of the encoding APIs was measured regarding the message size through a network sniffer. Figure 3 plots the encoding message size for each of the alarm message supporter by the system in the three message encoding APIs under test.

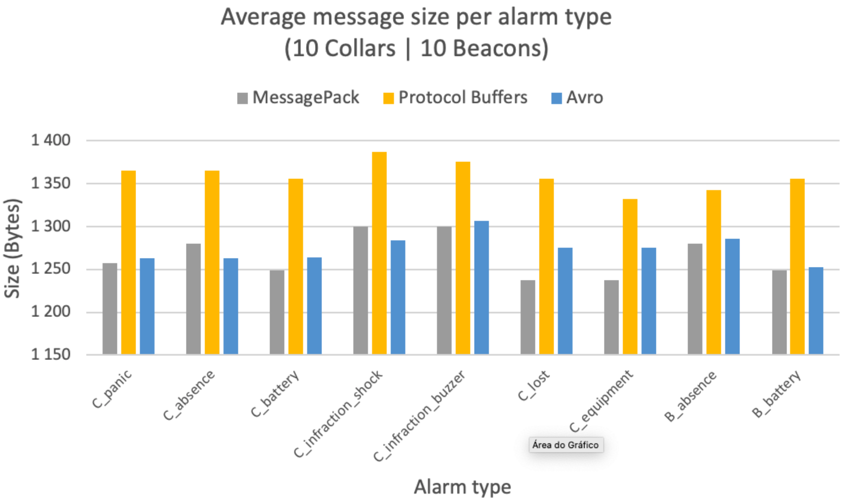

Figure 3. Analysis of the size of generated information for different alarms and encoding APIs

The test of the previous section revealed that MessagePack was the adequate format to encode the information forwarded to the cloud-hosted application. Moreover, experiences show that the evolution of the system's latency in alarm message transmission using the different API follow the encoding efficiency. As a result, the alarm types that produce bigger messages also take longer to transfer the information.

Once verified the greater efficiency of the MessagePack API, the JSON encoding API was replaced by the MessagePack API and the tests for transferring gateway generated alarms were repeated. To assess the impact of these changes, two metrics were measured: the volume of traffic produced and the corresponding latency.

The test conditions were similar to initial tests in terms of number of alarms and in the distribution of alarm types, but the periodicity messaging was set at 60 seconds. These results were compared to the results obtained before the system's optimization, and they are presented in Figure 4. Not surprisingly, the alarm dispatching latency followed message volume for both encodings.

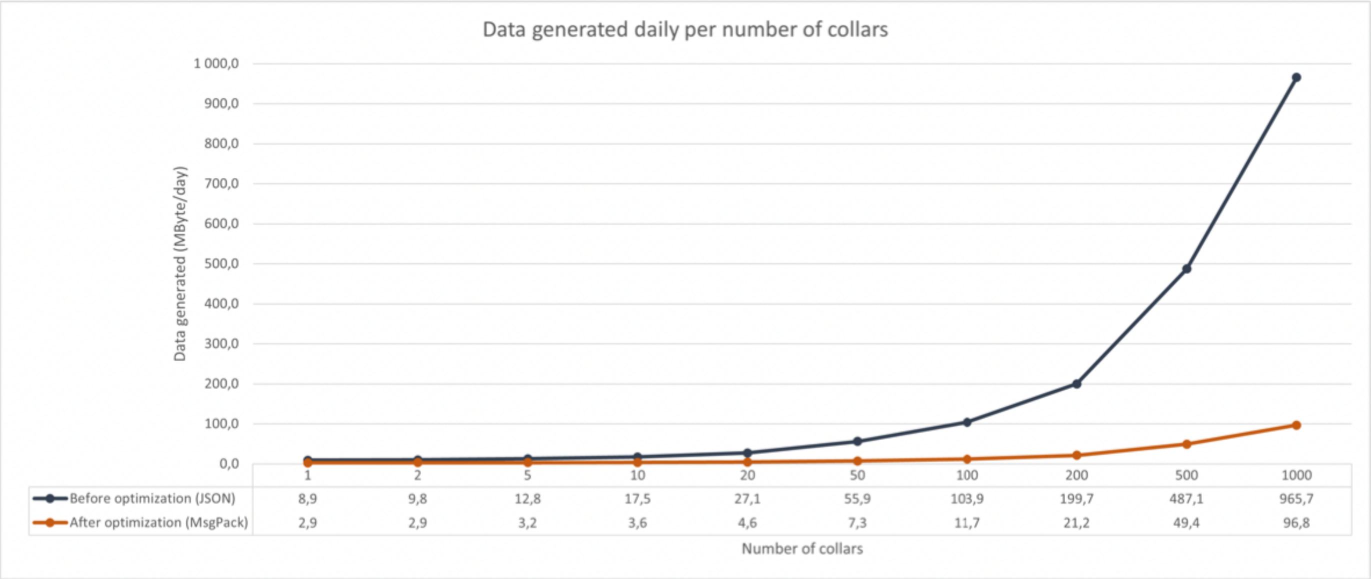

Figure 4. Comparison between original JSON format and MessagePack encoding efficiency.

3.3. *On-site tests*

To test the performance of the system in a real environment, we simulated sending events recorded by the gateway during the monitoring period of a flock of 18 sheep between the 18th and 29th of November 2021, and Figure 5 shows the number of alarms that were generated for the days observed. Since most of the alarms are associated with the behavior of living beings, the number of alarms generated daily does not follow any regular pattern. Furthermore, the number of alarms reached its maximum value on November 24th, when 192 alarms were detected. Approximately 44% of the daily values are in the range of 100, and 140 and 22% of the days generated less than 10 alarms. Moreover, the system registered an average of, approximately, 100 alarms per day.

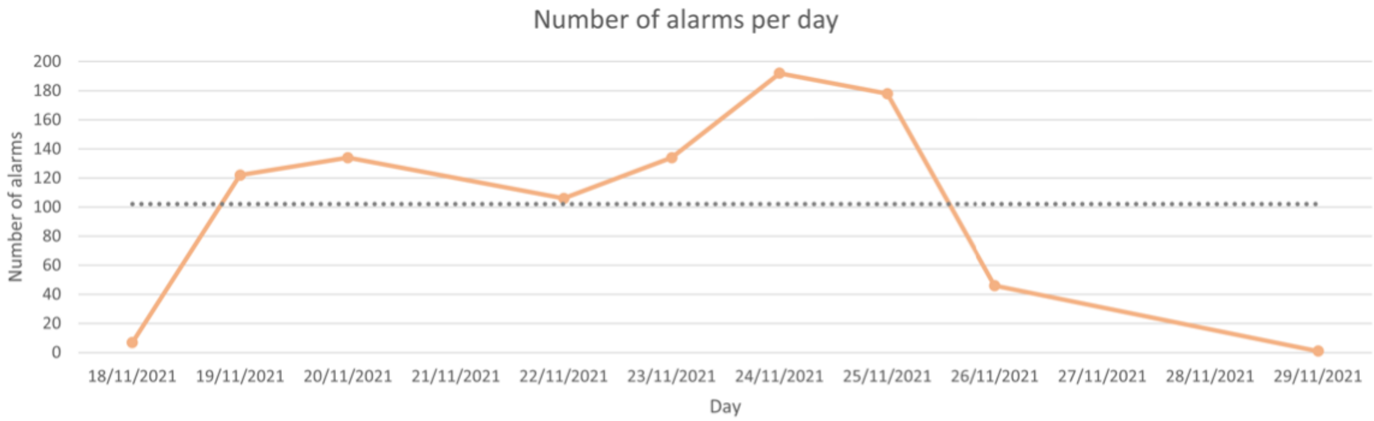

Figure 5 - Variation of the number of alarms throughout the 12 days.

Table 3 displays the number of alarms and their sizes, grouped by alarm type, that were produced during those days. Some alarm types have greater values than the expected (in a realistic scenario), which is explained by the characteristics of the system in question: *(i)* the infraction alarms were produced despite the conditioning being disabled, thus, the animals were not warned when committing an undesirable behavior and continued to have the same actions, triggering these alarms repeatedly; *(ii)* the beacon did not cover the entire pasture area, which means that some animals were not being detected even though they were inside the designated area.

Table 3. Number of alarms generated for alarm type.

Device	**Alarm type**		**Size (Bytes)**
Collar	Panic	6	1159
	Absence	75	1147
	Battery	2	1137
	Infraction	689	1126
	Lost	0	1134
	Equipment	141	1157
Beacon	Absence	3	1159
	Battery	4	1134

To validate the possibility that the hardware used in the gateway did not compromise performance, the alarm detection and sending times were measured, both for the gateway implemented on a Raspberry Pi and on a Core i7 4500U (1.8 GHz), 16 GB RAM PC.

As for the temporal analysis results illustrated in Figure 6, the computer presents a greater performance than the Raspberry Pi in every test, which was expected due to its higher processing capacity.

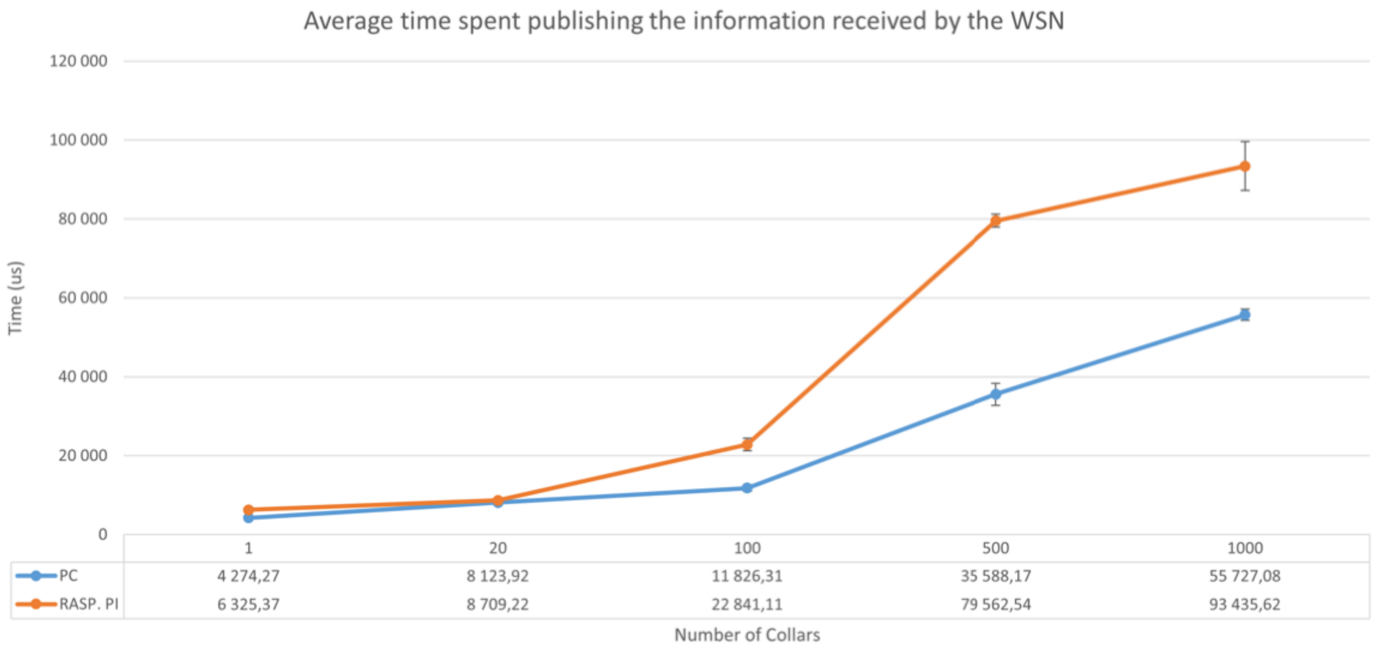

Figure 6. Impact of the number of collars on the time spent publishing the information received by the WSN.

4 Discussion

The results of the first test show an increase in the amount of data generated with the increase in number of collars and the decrease of the period (Figure 2), with a daily volume of 1.89 GB for 1000 collars and a period of 30 seconds. Although this volume of information can be transferred by the technology, which can transfer up to 2.99 GB daily, it has a quite an expensive cost, with daily values between 345 and 690€. These considerably expensive values would be even greater for larger herds as they are directly proportional to the values of generated data, displayed in Figure 2. As such, the increase of the data transfer period is not a viable solution, since the volume of data produced continues to represent a great expense for the system, therefore a message encoding optimization is required.

The results of the connection latency test evidence a higher value for the satellite case, associating as well that increase with the messages' size. When smaller messages are considered, the latency is also impacted by the different data path used by the two technologies (as discussed in the next paragraph). When larger messages are used, overall latency reached 1.5 seconds, which is much higher than the necessary in the Wi-Fi connection, which is due to the bandwidth available for the satellite connection.

Moreover, the satellite connection presents a considerably greater latency time than the WiFi, due to two major factors. Firstly, the path taken by the messages is not the same in each technology. The information sent via the satellite link travels to the destination using a greater number of hops; 23 hops in the case of satellite connection and 9 in the case of WiFi connection, were verified. Lastly, the data rate available for the satellite connection is limited, and thus, the same information takes more time to be uploaded in the case of this connection. Even so, the latencies of satellite communications cannot be considered a limiting factor for the system.

The encoding performance comparison results shown in Figure 3 consistently evidence higher efficiency of the MessagePack API for carrying alarm messages. AVRO encoding performed the second in the tests, and Protocol Buffers had the worst performance in test. The difference between MessagePack and Avro was systematically small, and between MessagePack and Protocol Buffers slightly larger, with about 8% difference.

The signaling traffic volume generated by the system, before and after message encoding optimization, considerably decreased and allowed a significant reduction on the amount of information exchanged daily, with approximately 10 times less data from the 100 collars upwards. Considering a tariff based on the information volume, the optimization has an impact considerable impact both on cost and on the system latency. The modification of the message encoding allowed a significant reduction on the amount of information exchanged daily, with approximately 10 times less data from the 100 collars upwards. In terms of latency, it reached to 4 seconds for the maximum number of collars tested and for 100 collars it decreased to, approximately, 2.9 times less.

The analysis of the computational requirements of the alarm system showed that for small and medium-sized herds, the hardware currently running at the gateway (RaspBerry PI) responds to the set of requests imposed by system events, but that for larger herds, with more than 100 animals, the response times grow too large to allow timely alarms to be sent. Not surprisingly, in the case of the gateway implemented via a PC, the times grew much more slowly, which makes it possible to create a gateway solution with better performance for larger herds.

5 Conclusions

An electronic grazing system allows the shepherd to be released from an arduous and lonely task, while continuing to monitor the animals closely and precisely. However, it still depends on the action of the human operator in certain situations, such as when the equipment fails or when the animals' wellbeing is at stake. The present work allowed for the implementation of a system that monitors animals devices, sending a set of alarms, to guarantee the safety of the vineyards and the animals.

Most vineyards are in rural areas, like the vines of SheepIT use case, and therefore have poor cellular coverage, hassling WSN connection to the Internet. As such, SheepIT gateway was integrated with a satellite interface, allowing it to operate even in places without Internet, such as the uneven slopes of the areas near the Douro river region in Portugal. In addition to the integration with the satellite communications system, the present work included the evaluation of several information encoding APIs, to optimize the sending of information related to alarms, and the replacement of the JSON API initially used by the MessagePack API.

The monitoring system was functionally validated, and they were evaluated the communication latency and it was measured the volume of signaling produced during its operation with three information encoding APIs. Signaling measurement revealed that MessagePack allowed better performance. Moreover, results shown that the performance values are perfectly acceptable and compatible with the system alarm function.

Additionally, the analysis of the computational requirements necessary for the implementation of the gateway showed that for small/medium sized herds it is possible to maintain a gateway based on the RaspBerry Pi, but that for larger sized herds it will be necessary to improve the computational capacity of the hardware that implements the gateway.

As future work, a LoRaWAN-satellite integrated environment, supported with AI-assisted opportunistic transmission will be considered, that shall allow timely alarm transmission, and opportunistically transmit at a low-cost the monitoring data for behavior analysis through data mining techniques.

Author Contributions: Pedro Gonçalves and Daniel Corujo were responsible for the Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision, project administration, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020-UIDP/50008/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Acknowledgments: We would like to express as well our thanks to EchoStar Mobile (https://www.echostarmobile.com/) for allowing us access to their satellite terminal and mobile satellite data services.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Banhazi, T.M.; Babinszky, L.; Halas, V.; Tscharke, M. Precision Livestock Farming: Precision Feeding Technologies and Sustainable Livestock Production. *International Journal of Agricultural and Biological Engineering* **2012**, *5*, 54–61, doi:10.25165/ijabe.v5i4.600.
2. Terrasson, G.; Villeneuve, E.; Pilnière, V.; Llaria, A. Precision Livestock Farming: A Multidisciplinary Paradigm. In Proceedings of the SMART 2017, The Sixth International Conference on Smart Cities, Systems, Devices and Technologies; IARIA, Ed.; ThinkMind: Venice, Italy, 2017; p. 55 to 59.

3. Nóbrega, L.; Pedreiras, P.; Gonçalves, P.; Nóbrega, L.; Pedreiras, P.; Gonçalves, P. SheepIT - An Electronic Shepherd for the Vineyards. In Proceedings of the 8th International Conference on Information and Communication Technologies in Agriculture, Food & Environment; Chania, Crete, Crete, Greece, 2017.
4. Gonçalves, P.; Nóbrega, L.; Monteiro, A.; Pedreiras, P.; Rodrigues, P.; Esteves, F. SheepIT, an E-Shepherd System for Weed Control in Vineyards: Experimental Results and Lessons Learned. *Animals* **2021**, *11*, 2625, doi:10.3390/ani11092625.
5. Kaler, J.; Mitsch, J.; Vázquez-Diosdado, J.A.; Bollard, N.; Dottorini, T.; Ellis, K.A. Automated Detection of Lameness in Sheep Using Machine Learning Approaches: Novel Insights into Behavioural Differences among Lame and Non-Lame Sheep. *R Soc Open Sci* **2020**, *7*, doi:10.1098/RSOS.190824.
6. Williams, M.; Davis, C.N.; Jones, D.L.; Davies, E.S.; Vasina, P.; Cutress, D.; Rose, M.T.; Jones, R.A.; Williams, H.W. Lying Behaviour of Housed and Outdoor-Managed Pregnant Sheep. *Appl Anim Behav Sci* **2021**, *241*, 105370, doi:10.1016/j.applanim.2021.105370.
7. Kleanthous, N.; Hussain, A.; Khan, W.; Sneddon, J.; Liatsis, P. Deep Transfer Learning in Sheep Activity Recognition Using Accelerometer Data. *Expert Syst Appl* **2022**, *207*, 117925, doi:10.1016/J.ESWA.2022.117925.
8. Gonçalves, P.; Marques, M.R.; Belo, A.T.; Monteiro, A.; Braz, F. Goat Kidding Dataset. *Data (Basel)* **2022**, *7*, 89, doi:10.3390/data7070089.
9. Rainho, R.; Corujo, D.; Gonçalves, P. SpaceSheep: Satellite-Aided E-Shepherd System.
10. Nóbrega, L.; Gonçalves, P.; Pedreiras, P.; Pereira, J. An IoT-Based Solution for Intelligent Farming. *Sensors* **2019**, *19*, 603, doi:10.3390/s19030603.
11. Kim, K. Mountainous Terrain Coverage in Mobile Sensor Networks. *IET Communications* **2015**, *9*, doi:10.1049/iet-com.2014.0443.
12. Temprilho, A.; Nóbrega, L.; Gonçalves, P.; Pedreiras, P.; Silva, S. M2M Communication Stack for Intelligent Farming. In Proceedings of the 2018 Global Internet of Things Summit; Bilbao, Spain, 2018.
13. Beyene, Y.D.; Jantti, R.; Tirkkonen, O.; Ruttik, K.; Iraji, S.; Larmo, A.; Tirronen, T.; Torsner, J. NB-IoT Technology Overview and Experience from Cloud-RAN Implementation. *IEEE Wirel Commun* **2017**, *24*, 26–32, doi:10.1109/MWC.2017.1600418.
14. Gomez, C.; Veras, J.C.; Vidal, R.; Casals, L.; Paradells, J. A Sigfox Energy Consumption Model. *Sensors 2019, Vol. 19, Page 681* **2019**, *19*, 681, doi:10.3390/S19030681.
15. Liya, M.L.; Arjun, D. A Survey of LPWAN Technology in Agricultural Field. *Proceedings of the 4th International Conference on IoT in Social, Mobile, Analytics and Cloud, ISMAC 2020* **2020**, 313–317, doi:10.1109/I-SMAC49090.2020.9243410.
16. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment. *ICT Express* **2019**, *5*, 1–7, doi:10.1016/j.icte.2017.12.005.
17. Stankovic, J.A. Wireless Sensor Networks. *Computer (Long Beach Calif)* **2008**, *41*, 92–95, doi:10.1109/MC.2008.441.
18. Nóbrega, L.; Gonçalves, P.; Antunes, M.; Corujo, D. Assessing Sheep Behavior through Low-Power Microcontrollers in Smart Agriculture Scenarios. *Comput Electron Agric* **2020**, *173*, 105444, doi:https://doi.org/10.1016/j.compag.2020.105444.
19. Guedes, R.; Pedreiras, P.; Nóbrega, L.; Gonçalves, P. Towards a Low-Cost Localization System for Small Ruminants. *Comput Electron Agric* **2021**, *185*, 106172, doi:10.1016/j.compag.2021.106172.
20. Andresen, T.; De Aguiar, F.B.; Curado, M.J. The Alto Douro Wine Region Greenway. *Landsc Urban Plan* **2004**, *68*, doi:10.1016/S0169-2046(03)00156-7.
21. Aqui Fica a Saber Available online: https://anacom.maps.arcgis.com/apps/Cascade/index.html?appid=ad3f71dbb09541518f436aa828feb28e (accessed on 4 April 2023).

22. Cabrera-Castellanos, D.F.; Aragón-Zavala, A.; Castañón-Ávila, G. Closing Connectivity Gap: An Overview of Mobile Coverage Solutions for Not-Spots in Rural Zones. *Sensors* **2021**, *21*, 8037, doi:10.3390/s21238037.
23. Crockford, D. The Application/Json Media Type for JavaScript Object Notation (JSON). **2006**, doi:10.17487/RFC4627.
24. Rostanski, M.; Grochla, K.; Seman, A. Evaluation of Highly Available and Fault-Tolerant Middleware Clustered Architectures Using RabbitMQ. In Proceedings of the Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on; IEEE, 2014; pp. 879–884.
25. Vohra, D. Apache Avro. *Practical Hadoop Ecosystem* **2016**, 303–323, doi:10.1007/978-1-4842-2199-0_7.
26. Travers Ching, by; Eddelbuettel, D. RcppMsgPack: MessagePack Headers and Interface Functions for R.
27. Hughes 4200 Portable Data Terminal - EchoStar Mobile Available online: https://www.echostarmobile.com/satellite-terminals/hughes-4200-portable-data-terminal/ (accessed on 5 April 2023).