Security and storage issues in Internet of Floating Things edge-cloud data movement

Montella, Raffaele; Di Luccio, Diana; Kosta, Sokol; Castiglione, Aniello; Maratea, Antonio

Published in:
Parallel Processing and Applied Mathematics - 13th International Conference, PPAM 2019, Revised Selected Papers

DOI (link to publication from Publisher):
10.1007/978-3-030-43222-5_10

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Security and storage issues in Internet of Floating Things edge-cloud data movement

Raffaele Montella[1], Diana Di Luccio[1], Sokol Kosta[2], Aniello Castiglione[1], and Antonio Maratea[1]

[1] Department of Science and Technologies, University of Naples Parthenope, Naples, Italy
{raffaele.montella,diana.diluccio,
aniello.castiglione,antonio.maratea}@uniparthenope.it
[2] Department of Electronic Systems, Aalborg University, Copenhagen, Denmark
sok@es.aau.dk

Abstract. Sensors and actuators became first class citizens in technologically pervasive urban environments. However, the full potential of data crowdsourcing is still unexploited in marine coastal areas, due to the challenging operational conditions, extremely unstable network connectivity and security issues in data movement. In this paper, we present the latest specification of our DYNAMO Transfer Protocol (DTP), a platform-independent data mover framework specifically designed for the Internet of Floating Things applications, where data collected on board of professional or leisure vessels are stored locally and then moved from the edge to the cloud. We evaluate the performance achieved by the DTP in data movement in a controlled environment.

Keywords: Internet of Floating Things · Data crowdsourcing · Data Movement · Security · Cloud Database.

1 Introduction

The rise of the Internet of Things and the computational resource elasticity provided by the Cloud [33,6] are changing the human lifestyle [9]. The crowdsourcing paradigm [14] has been widely adopted in diverse contexts to solve large problems by engaging many human workers to solve manageable sub-problems [12]. When the problem involves data acquisition, management and analysis, it is referred to as *data crowdsourcing* [12]. Nowadays, data crowdsourcing is one of the most impacting technology raised as first-class citizen in the data science landscape [32], thanks to the flywheel effect generated by the availability of distributed human-carried sensor network – commonly referred as mobile computing, the reliable connection infrastructure provided by cellular and Wi-Fi networks, and the elastic computing and storage resources. Nevertheless, data crowdsourcing potentially gains more importance in environments where the use of conventional data acquisition methodologies [27,2,17,20] are expansive or unfeasible and the satellite data do not reach the adequate resolution and quality,

mostly when approaching the coast [3,4]. The coastal areas host most part of human population, are fundamental for the global economy, and, above all, are one of the most sensitive environments to climate changes [26]: extreme weather events could impact negatively on human activities in a dramatic way [30,7].

Although the embryo of a distributed data collecting has been already designed at the early stage of the grid computing epic [10,34], unfortunately, data crowdsourcing marine applications are limited by the availability of stable, reliable, and cheap data connections. On the other hand, the measurement of seafloor bottom depth (bathymetry) via data crowdsourcing is common in both scientific [29] and business projects. The application of this technique to the measurement of weather and sea state parameters has previously been limited to ferries, freight carriers, professional vessels, and cruise ships. In a previous work, we developed FairWind, a smart, cloud-enabled, multi-functional navigation system for leisure and professional vessels [22,25]. In this paper, we introduce DYNAMO, an infrastructure for collecting marine environmental data from a distributed sensor network carried by leisure vessels [24,21]. DYNAMO could be considered as an improvement and evolution of FairWind, strongly leveraging on SignalK (`http://signalk.org`) as a common interchange format for marine data, but more focused on data logging and management.

The rest of this paper is organized as follows: Section 2 contains a synthetic description of similar solutions focusing on diverse and different data transfer protocols. Section 3 contains the most of the novel contribution of this paper with a detailed description of the DYNAMO Transfer Protocol, detailing on the security and storage issues. Section 4 describes the preliminary evaluation in an experimental controlled environment. Finally, Section 5 concludes and outlines future directions.

2 Related Work

The Bundle Protocol (BP) [28] has been proposed by the Delay Tolerant Networking Research Group (DTNRG) of the Internet Research Task Force (IRTF). The idea of this protocol is to group data in bundles in order to store and forward them when the networking is available. The main capabilities of the BP include: *i)* custody-based re-transmission; *ii)* ability to cope with intermittent connectivity; *iii)* ability to take advantage of scheduled, predicted, and opportunistic connectivity; *iv)* late binding of overlay network endpoint identifiers to constituent internet addresses. Even though BP is the only acknowledged data transfer protocol for DTNs as of today, and the best reference point for new proposals in this field, it is not designed for IoT devices and for their communication with cloud infrastructures.

The two widely used IoT application protocols that represent the current state of the art, are Message Queuing Telemetry Transport (MQTT) [15] and Constrained Application Protocol (CoAP) [5]. MQTT is an internet application protocol for extreme environments. MQTT today is an OASIS standard, widely used for every kind of IoT application, including cloud data transfer. It is a

publish/subscribe model working on top of TCP, ensuring the reliability of its approach also thanks to its small bandwidth footprint and a low loss rate in unstable networking.

CoAP is a modern standard specialized application protocol for constrained devices. It leverages on a REST model: servers allow resources access under a URL and clients use them through GET, POST, PUT, and DELETE methods. This makes CoAP integration with already different software straightforward, but working on UDP in order to maximize the efficiency.

A common middleware supporting both MQTT and CoAP and providing a common programming interface has been also implemented [31].

Nevertheless, MQTT and CoAP are both not resilient, since they are not able to elastically change the transmission rate in dependence of the bandwidth, and although both have a lightweight footprint, they do not compress the payload. The security is guaranteed by the transport layer not ensuring the firewall and proxy friendship.

3 Design

3.1 Vessel side and cloud side security

In [25], we already described the idea of a data transfer framework designed for vessel data logging and transferring to the cloud with an adaptive algorithm devoted to the maximization of the bandwidth usage leveraging on concurrent requests. In this work, we completely redesigned the vessel side in order to match a higher level of security avoiding any form of key exchange. The behaviour of the SignalK data logger on the vessel side is conceptually described by the block diagram shown in the Figure 1.

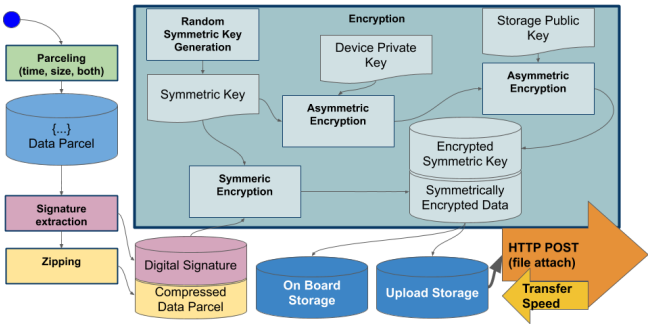

Fig. 1. The DYNAMO SignalK Logger on the vessel side.

The SignalK data updates are grouped in parcels, as described in [21], and stored as text files in a temporary outbound folder. Each time a new par-

cel is available, the signature is extracted using the RSA-SHA 256 bit algorithm, encrypted using the vessel 1024 bit private key and finally encoded in base64 [11]. A 32 byte long symmetric encryption key is generated randomly. This key is encrypted with the cloud-side public key using the RSA PLCS1 OAEP padding. The encrypted symmetric key is finally encoded in base 64. A cipher key is generated using the SHA256 hash algorithm applied to the previously generated symmetric key. A secure pseudo random initialization vector is generated and then used to encrypt the compressed data parcel. Finally, the initialization vector and the encrypted symmetric key are prepended and the encrypted data parcel is stored in an upload folder. The DYNAMO data transfer framework mediates the vessel and the cloud sides in order to maximize the bandwidth usage. A NodeJS working implementation of a SignalK DYNAMO logger is available as open source (Apache 2.0 license)(`https://github.com/OpenFairWind/signalk-dynamo-logger`).

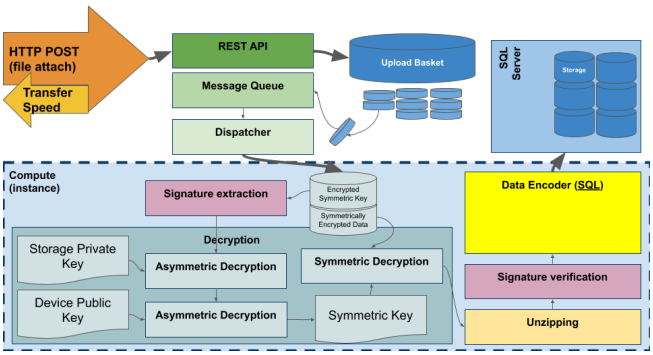

Fig. 2. *Cloud storage side* (consumer) data pipeline. The symmetric key is decrypted with the consumer's private key, and then is used to decrypt the data. The data are then uncompressed, the signature is verified, and the data are stored in a SQL database.

The cloud storage side is described in Figure 2. Each time an encrypted and compressed data parcel is received, it is stored in an upload basket and enqueued to a message queue manager. The behavior of this component drastically affects the overall storage performance, which makes it a critical point on this kind of near-realtime applications. In order to be as much as possible independent from the practical implementation choices, we used a a plug-in approach enabling the DYNAMO cloud storage administrator to change the message queue manager and its policies in order to match the specific application and, above all, the available storage and computational resources. For production scenarios the use of enterprise level components as RabbitMQ (`https://www.rabbitmq.com`) and Redis (`https://redis.io`) are recommended, but for performance evaluation we implemented our own message queue manager in order to make the metric

measurements and the control over the used resources more effective [16]. The message queue manager dispatches the encrypted and compressed data parcels on available computational resources. Here the encoded encrypted symmetric key is extracted and decrypted using the cloud side private key in order to enforce the non repudiability of the data. Each data parcel contains a list of SignalK updates. The symmetric key is used to decrypt the compressed data parcel using the initialization vector leveraging on the AES algorithm using the CBC mode. Then, the data parcel is unzipped and the encrypted digital signature extracted and decrypted using the vessel public key. Finally, the decrypted digital signature is used for verification in order to enforce the integrity, then the updates are added to the update list. Once the update list is fully consistent, the storage in database process begins. A Python working implementation of a SignalK DYNAMO cloud storage is available as open source (Apache 2.0 license - `https://github.com/OpenFairWind/dynamo-storage`).

3.2 Cloud side storage

At the cost of a time-consuming, careful and proper data schematization, the relational solution offers many advantages over the plain NoSQL one, namely: i) a highly expressive manipulation language (SQL); ii) minimal redundancy; iii) the possibility to enforce sophisticated integrity constraints; iv) indexing, materialization, partitioning and all the arsenal of physical optimization to improve performance. In this case, the main challenge for designing an E/R diagram is the unknown schema of the received data, that concern different aspects of the vessel navigation gathered in real time and in a semi-structured form. Furthermore, all the measured quantities evolve over time and are sampled at irregular intervals, due to the possibly harsh environmental conditions and the consequent loss of signals.

The proposed solution is a *star schema*, with a strong entity at the center representing the VESSEL (the transmitters) and a variable number of weak entities at its side that hold the data relative to each variable to be stored, arranged time-wise (the TIMESTAMP of the measurements is the weak key for all the weak entities). For example, a variable like "position" is measured and transmitted ideally with a 5 to 10 Hz frequency and stored in a table named POSITION whose primary key is the combination of VESSEL-ID and TIMESTAMP and whose attributes are LATITUDE and LONGITUDE; a variable like "destination" is measured and transmitted with a given frequency and stored in a table named DESTINATION whose primary key is the combination of VESSEL-ID and TIMESTAMP and whose attributes are characteristics of the destination, like DESTINATION-COMMON-NAME. In both tables, the VESSEL-ID is also a foreign key connecting the weak entity to the strong entity (the VESSEL table). Once data relative to POSITION (or to DESTINATION) arrive with attached the TIMESTAMP of the measurements, they are stored in the corresponding tables. This schema naturally partitions the load on the tables, allowing parallel inserts.

The schema is built incrementally and dynamically as new variables arrive from the floating things, becoming new tables. After a *boot* period during which

many new tables (one for each unknown variable) are created, the schema stabilizes itself and variable addition becomes very rare. All consecutive measurements of the same variable are stored in the corresponding table sorted by time of measurement (TIMESTAMP). This schema does not require the *a priori* definition of the number or type of required variables, allows enforcing of integrity constraints, holds the time series of each variable and can be indexed and tuned via standard SQL to improve the access time.

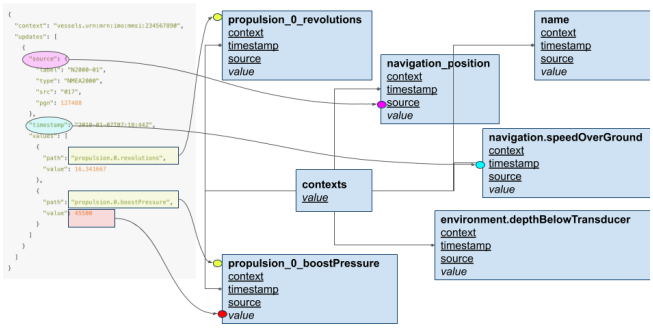

Fig. 3. SQL data encoder.

In the Internet of Floating Things, all transmitters use the GPS time, hence they are all synchronized, however, depending on the application, a time quantization is normally necessary to have a meaningful match of the tuples representing the measurements of the variables on the various tables. If we define ϵ the minimum time interval considered relevant for the application (for example $\epsilon = 1s$), all values of a given variable in the same time windows (1 second) should be averaged and only their average value (or a more robust index [18]) should be considered, with a timestamp truncated to the second. Quantization can be performed adding to each table a column QUANT representing the time as the number of ϵ unit of times (seconds) passed since a reference date and then grouping.

In general, there are two possible choices: a) storing the raw data at maximum time resolution and performing the quantization *after* inserting the data in the database, querying and joining them with GROUP BY clauses on the QUANT column, eventually saved in a materialized view; b) using the GROUP BY clauses on the QUANT column to quantize and join the data *before* inserting them in the database, averaging their values.

The advantage of solution (a) is that the granularity can be changed at the application level, while in solution (b) the raw data are lost and the data can only be rolled up. On the other hand, solution (a) requires more space and more write operations with respect to solution (b), that is lighter and may give a

sufficient precision in most real use cases. In the following we adopted solution (a).

4 Evaluation

In order to produce a preliminary evaluation of the implemented DYNAMO cloud storage, we set up an experiment using the HPC cluster *PurpleJeans* available at the Department of Science and Technologies of the University of Naples Parthenope as controlled environment. The cluster is devoted to Machine Learning and Data Science researches. Although the cluster is provided by a 16 NVidia V100 CUDA enabled GPGPU devices partition, we used the multicore intensive computing partition powered by 4 computing nodes equipped by 2 Intel Xeon 16-Core 5218 2,3Ghz CPUs providing 32 computing cores per node and supported by 192 Giga Bytes of RAM. The computing nodes are connected to the front-end with an Infiniband Mellanox CX4 VPI SinglePort FDR IB 56Gb/s x16. The file system is shared using the Ethernet over the Infiniband protocol. The cluster supports Docker on both front-end and computing nodes. The total amount of storage is about 65 Terabytes. We configured the DYNAMO Cloud Storage using a custom message queue manager, dispatching the execution of the decryption and decompressing tasks on Docker-deployed DYNAMO cloud storage compute instances on the computing nodes. Using Docker, we deployed a single instance of PostgreSQL/PostGIS SQL database server on the front-end (Figure 4). We simulated a workload with 5182 encrypted and compressed data parcels acquired during a real vessel navigation sequentially enqueued to the message queue manager. The used dataset produces 21 tables in the SQL database. We performed the overall wall clock measurement varying the number of deployed DYNAMO cloud storage compute instances on the computing nodes. All containers share the same Docker volume and they can interact with the SQL database server instance. The preliminary results obtained performing 100 times the described experiments are shown in the right side of the Figure 4. Under the described experiment setup, the proposed methodology scales almost linearly up to 16 DYNAMO cloud storage instances as expected. This is also supported by the number of SQL tables automatically generated that is less or almost equal to the number of the compute instances. In this way, the SQL server lock is at the table level and multiple insert queries could be executed concurrently on the single SQL database server instance. As the number of instances increases, the performance gains no benefits from the concurrent inserts.

5 Conclusion and future directions

In this paper, we presented the latest implementation of the DYNAMO Transfer Protocol, which enforces a complete end to end encryption methodology with data signature in order to ensure data integrity, non repudiability and, above all, privacy, since the data in this context are related to people and goods and contain position and time references. The DYNAMO ecosystem could contribute to the

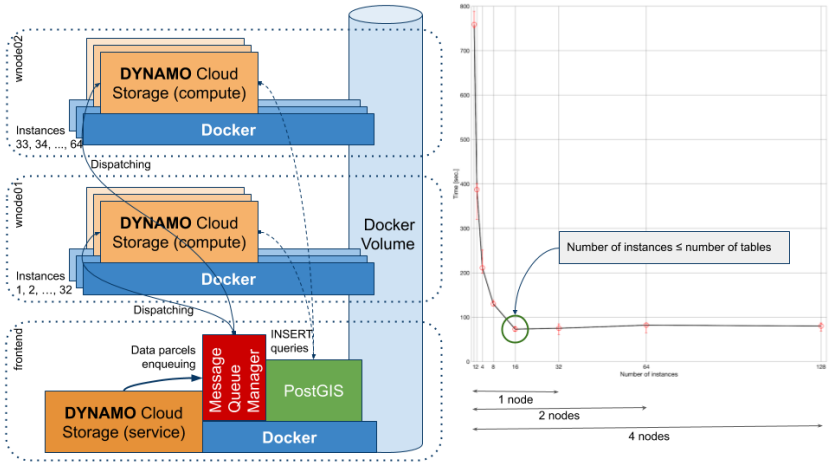

Fig. 4. The experiment setup (left) and the preliminary results (right) of the DYNAMO cloud storage.

search and rescue system: in designing the proposed framework, we considered the IoT security aspects [34] in order to avoid disruptive situations that could affect the safety at sea. All these elements lead us to believe that the DYNAMO ecosystem will gain robustness and effectiveness using the proposed data transfer approach.

Leveraging on a more sophisticated parallelization techniques [8], the final goal is building a progressively improving dataset about coastal marine environmental data [19] with a twofold utilization *i)* training the next generation of deep learning models in order to carry out useful information for strategical resources management and providing assimilation data for predicting and simulating models [13,1] and workflows [23].

Acknowledgment

This research was supported by the research project "DYNAMO: Distributed leisure Yacht-carried sensor-Network for Atmosphere and Marine data crOwdsourcing applications" (DSTE373) and it is partially included in the framework of the project "MOQAP - Maritime Operation Quality Assurance Platform" and financed by Italian Ministry of Economic Development.

References

1. Ascione, I., Giunta, G., Mariani, P., Montella, R., Riccio, A.: A grid computing based virtual laboratory for environmental simulations. In: European Conference on Parallel Processing. pp. 1085–1094. Springer (2006)

2. Aulicino, G., Cotroneo, Y., Ansorge, I., Berg, M.v.d., Cesarano, C., Belmonte Rivas, M., Olmedo Casal, E.: Sea surface salinity and temperature in the southern atlantic ocean from south african icebreakers, 2010–2017. Earth System Science Data **10**(3), 1227–1236 (2018)
3. Benassai, G., Di Luccio, D., Corcione, V., Nunziata, F., Migliaccio, M.: Marine spatial planning using high-resolution synthetic aperture radar measurements. IEEE Journal of Oceanic Engineering **43**(3), 586–594 (2018)
4. Benassai, G., Di Luccio, D., Migliaccio, M., Cordone, V., Budillon, G., Montella, R.: High resolution remote sensing data for environmental modelling: Some case studies. In: 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI). pp. 1–5. IEEE (2017)
5. Bormann, C., Castellani, A.P., Shelby, Z.: Coap: An application protocol for billions of tiny internet nodes. IEEE Internet Computing **16**(2), 62–67 (2012)
6. Botta, A., De Donato, W., Persico, V., Pescapé, A.: On the integration of cloud computing and internet of things. In: Future internet of things and cloud (FiCloud), 2014 international conference on. pp. 23–30. IEEE (2014)
7. Di Paola, G., Aucelli, P.P.C., Benassai, G., Rodríguez, G.: Coastal vulnerability to wave storms of sele littoral plain (southern italy). Natural hazards **71**(3), 1795–1819 (2014)
8. DAmore, L., Mele, V., Laccetti, G., Murli, A.: Mathematical approach to the performance evaluation of matrix multiply algorithm. In: Parallel Processing and Applied Mathematics, pp. 25–34. Springer (2016)
9. Fortino, G., Trunfio, P.: Internet of things based on smart objects: Technology, middleware and applications. Springer (2014)
10. Foster, I.: Globus online: Accelerating and democratizing science through cloud-based services. IEEE Internet Computing **15**(3), 70–73 (2011)
11. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: Rsa-oaep is secure under the rsa assumption. In: Annual International Cryptology Conference. pp. 260–274. Springer (2001)
12. Garcia-Molina, H., Joglekar, M., Marcus, A., Parameswaran, A., Verroios, V.: Challenges in data crowdsourcing. IEEE Transactions on Knowledge and Data Engineering **28**(4), 901–911 (2016)
13. Giunta, G., Montella, R., Mariani, P., Riccio, A.: Modeling and computational issues for air/water quality problems: A grid computing approach. Nuovo Cimento C Geophysics Space Physics C **28**, 215 (2005)
14. Heipke, C.: Crowdsourcing geospatial data. ISPRS Journal of Photogrammetry and Remote Sensing **65**(6), 550–557 (2010)
15. Hunkeler, U., Truong, H.L., Stanford-Clark, A.: Mqtt-sa publish/subscribe protocol for wireless sensor networks. In: 3rd international conference on Communication systems software and middleware and workshops. pp. 791–798. IEEE (2008)
16. Laccetti, G., Lapegna, M., Mele, V.: A loosely coordinated model for heap-based priority queues in multicore environments. International Journal of Parallel Programming **44**(4), 901–921 (2016)
17. Mangoni, O., Saggiomo, V., Bolinesi, F., Margiotta, F., Budillon, G., Cotroneo, Y., Misic, C., Rivaro, P., Saggiomo, M.: Phytoplankton blooms during austral summer in the ross sea, antarctica: Driving factors and trophic implications. PLoS One **12**(4), e0176033 (2017)
18. Maratea, A., Gaglione, S., Angrisano, A., Salvi, G., Nunziata, A.: Non parametric and robust statistics for indoor distance estimation through ble. In: 2018 IEEE International Conference on Environmental Engineering (EE). pp. 1–6 (March 2018). https://doi.org/10.1109/EE1.2018.8385266

19. Marcellino, L., Montella, R., Kosta, S., Galletti, A., Di Luccio, D., Santopietro, V., Ruggieri, M., Lapegna, M., DAmore, L., Laccetti, G.: Using gpgpu accelerated interpolation algorithms for marine bathymetry processing with on-premises and cloud based computational resources. In: International Conference on Parallel Processing and Applied Mathematics. pp. 14–24. Springer (2017)
20. Misic, C., Harriague, A.C., Mangoni, O., Aulicino, G., Castagno, P., Cotroneo, Y.: Effects of physical constraints on the lability of pom during summer in the ross sea. Journal of Marine Systems **166**, 132–143 (2017)
21. Montella, R., Di Luccio, D., Kosta, S., Giunta, G., Foster, I.: Performance, resilience, and security in moving data from the fog to the cloud: The dynamo transfer framework approach. In: International Conference on Internet and Distributed Computing Systems. pp. 197–208. Springer (2018)
22. Montella, R., Di Luccio, D., Marcellino, L., Galletti, A., Kosta, S., Brizius, A., Foster, I.: Processing of crowd-sourced data from an internet of floating things. In: 12th Workshop on Workflows in Support of Large-Scale Science. p. 8. ACM (2017)
23. Montella, R., Di Luccio, D., Troiano, P., Riccio, A., Brizius, A., Foster, I.: Wacomm: A parallel water quality community model for pollutant transport and dispersion operational predictions. In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 717–724. IEEE (2016)
24. Montella, R., Kosta, S., Foster, I.: Dynamo: Distributed leisure yacht-carried sensor-network for atmosphere and marine data crowdsourcing applications. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). pp. 333–339. IEEE (2018)
25. Montella, R., Ruggieri, M., Kosta, S.: A fast, secure, reliable, and resilient data transfer framework for pervasive iot applications. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 710–715. IEEE (2018)
26. Paw, J.N., Thia-Eng, C.: Climate changes and sea level rise: implications on coastal area utilization and management in south-east asia. Ocean and Shoreline Management **15**(3), 205–232 (1991)
27. Rivaro, P., Ianni, C., Raimondi, L., Manno, C., Sandrini, S., Castagno, P., Cotroneo, Y., Falco, P.: Analysis of physical and biogeochemical control mechanisms on summertime surface carbonate system variability in the western ross sea (antarctica) using in situ and satellite data. Remote Sensing **11**(3), 238 (2019)
28. Scott, K.L., Burleigh, S.: Bundle protocol specification. RFC 5050, RFC Editor (November 2007), `https://tools.ietf.org/html/rfc5050`
29. Sedaghat, L., Hersey, J., McGuire, M.P.: Detecting spatio-temporal outliers in crowdsourced bathymetry data. In: 2nd ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information. pp. 55–62. ACM (2013)
30. Small, C., Gornitz, V., Cohen, J.E.: Coastal hazards and the global distribution of human population. Environmental Geosciences **7**(1), 3–12 (2000)
31. Thangavel, D., Ma, X., Valera, A., Tan, H.X., Tan, C.K.Y.: Performance evaluation of MQTT and CoAP via a common middleware. In: IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). pp. 1–6. IEEE (2014)
32. Turner, A.: Introduction to neogeography. " O'Reilly Media, Inc." (2006)
33. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications **1**(1), 7–18 (2010)
34. Zhou, J., Cao, Z., Dong, X., Vasilakos, A.V.: Security and privacy for cloud-based iot: Challenges. IEEE Communications Magazine **55**(1), 26–33 (2017)