Sparks-Edge: Analytics for Intelligent City Water Metering

Dimitrios Amaxilatis
Spark Works ITC Ltd.
Sheffield, United Kingdom
d.amaxilatis@sparkworks.net

Ioannis Chatzigiannakis
Sapienza University of Rome
Rome, Italy
ichatz@diag.uniroma1.it

Christos Tselios
ECE Department, University of Patras
Patras, Greece
tselios@ece.upatras.gr

Nikolaos Tsironis
Spark Works ITC Ltd.
Sheffield, United Kingdom
ntsironis@sparkworks.net

Abstract

Smart Meter infrastructures are emerging systems that measure, collect, and analyze utility data and communicate with the network's backbone on a fixed schedule. Such infrastructures are a vital part towards real Intelligent Cities. In this article we propose an edge-processing oriented Internet of Things architecture for smart meter networks that helps reduce data communication while keeping the system secure, reliable and responsive. We discuss our system architecture based on a real-world water metering deployment of 48 water meters inside a University Campus, using off-the-shelf wM-Bus water meters. We also provide a study of how our solution can face the same problems regardless of the size of the water meter network, scaling up to cities of millions of citizens and measuring points, reducing traffic and data sizes event by 80%.

1 Introduction

The advent of novel networking paradigms such as 5G and the Internet of Things (IoT) will lead to an exponential increase of interconnected devices, since everyday objects equipped with unique identifiers, will be capable of automatically connect to affiliated network interfaces and upload large volumes of highly diversified datasets. This rapidly-approaching, hyper-connected ecosystem aims to deliver an "always connected" end-user experience and will most probably need to augment all existing cloud computing deployments, which now struggle to handle the volume, the variety and the velocity of transmitted data streams. It is no coincidence that latency is constantly rising, often compromising delay-sensitive applications. It becomes obvious that for improving performance, decrease end-to-end over-the-air latency and boost availability and coverage, novel 5G features such as network slicing and more agile network architectures are now considered mandatory.

In: Proceeding of the Poster and Workshop Sessions of AmI-2019, the 2019 European Conference on Ambient Intelligence. Rome, Italy, November 2019, published at http://ceur-ws.org, Editors of the proceedings: Emilio Calvanese Strinati, Dimitris Charitos, Ioannis Chatzigiannakis, Paolo Ciampolini, Francesca Cuomo, Paolo Di Lorenzo, Damianos Gavalas, Sten Hanke, Andreas Komninos, Georgios Mylonas

Edge computing is a contemporary platform which deploys an intermediate layer of computational and storage resources paired with the necessary control functionality between end-user equipment and cloud computing datacenters. The edge infrastructure's physical proximity with the IoT sensors, greatly limits latency, decreases bandwidth consumption and delivers cutting-edge services of improved security and reliability. This approach extends the cloud computing paradigm by migrating data processing closer to production site, accelerates system responsiveness to events along with its overall awareness, by eliminating the data round-trip to the cloud. Offloading large datasets to the core network is no longer a necessity, consequently leading to improved safety and quality of experience (QoE) [13]. Moreover, the specific solution confronts several of the intrinsic limitations of cloud and alleviates the deployment of services with limited or even zero tolerance for error, such as Smart City monitoring.

Smart City-related applications become more and more common as well as pervasive, leading to an increased sensing node deployment density and network topology scale. Mostly depending on Low Power Wide Area Networking (LPWAN), an emerging network paradigm for IoT, Smart City monitoring infrastructure must remain low cost, energy efficient and capable of being widely deployed. Among all available LPWAN technologies, LoRa networking has attracted much attention from both academia and industry, since it specifies an open standard and allows the development of autonomous LPWAN networks, eliminating the necessity for proprietary hardware. This paper proposes a Smart City monitoring application which exploits specific characteristics of Edge computing and LoRa to provide a solution to address a real-world problem: water management misuse. This thorny issue is tackled through an intelligent platform that analyzes inbound water meter oriented datasets on the spot, while retaining an increased level of robustness and expandability.

2 Related Work

The IoT domain is always challenging due to the large potential number of sensor data that can be generated by an ever increasing number of sensing devices. Typically, the sensor devices are low-end but the idea of combining data close to their producers (i.e., in-network aggregation and on-the-spot data management) is considered a viable solution. The main advantage is more clear by the ability to combine heterogeneous datasets from multiple sources and with low latency, providing a better experience of end-users [13]. Such techniques are tightly bound with the lower-level medium access control protocols as well as network-level routing ones. Examples of such protocols are are presented in [7].

The arrival of Big Data solutions, with the help of the map-reduce technique provided us with multiple tools [5] (e.g., Apache Spark[1]) that simplify the process by splitting the data in distinct easily managed batches. Other tools, building on the map-reduce paradigm adopted a more streaming way of time-series analysis, resulting in Stream Processing Frameworks, with Apache Storm[2], Flink[3], and Heron[4] being the most common together with possible proposed optimizations [8]. Such solutions use the internal logic of the high-level application components [12] and are capable of confronting intrinsic cloud limitations thus alleviating the deployment of services with low or even zero tolerance for latency delays.

Complementary to sensor-originating traffic management and dataset handling, precise energy monitoring and conservation methods are aspects of great interest, mostly due to the imbalance between power generation and demand. Smart Grids [11] are an excellent playground for smart power meters that use advanced sensors and IoT-related technologies. An overlaying communication and information handling network like the Fog Computing paradigm can help progress the robustness and performance of monitoring frameworks. Residential monitoring prototypes for calculating and estimating domestic power consumption [10] have limited capabilities to find patterns using small-scale deployment data. Other low-price solutions[9] offer limited features and are totally lacking data manipulation and storage capabilities. [4] presents us with a more holistic approach integrated with structural building information from dedicated databases. It exploits recent advances in physical and environmental sensing together with digital repositories of buildings and districts. The prototype supports near-real-time energy consumption but lacks in scalability and process provisioning.

The notion of local data pre-processing to reduce data transfer between nodes was considered by [3, 2, 14]. This approach is more suitable for environments with limited data transfer capabilities and an intermediate layer of Fog Computing. The ever increased number of interconnected devices can inhibit the ability to transmit

[1]http://spark.apache.org/
[2]http://storm.apache.org/
[3]https://flink.apache.org/
[4]https://apache.github.io/incubator-heron/

datasets accross the Internet making it also significantly expensive. This is the main reason making our approach capable of exploiting local preprocessing removing the pains and shortages of both bandwidth and throughput faced by every network.

3 Architecture

The evaluation setup consists of 4 layers. The lower layer, contains a total of 44 off-the-shelf water consumption meters, 2 water pressure meters and 2 remote controlled valves deployed inside a University Campus. All the above broadcast their data using wM-Bus on predefined intervals (between 3 and 60 minutes depending on the device's configuration). The data reported include the total water consumption, the current water pressure, the water and environment temperature and the status of the water valve. Each message is encrypted individually and requires a separate (per meter) key to decrypt its data on the receiving end. The transmitted packets are collected by a network of 18 deployed wM-Bus-to-LoRaWAN bridges based on the STM32 Nucleo processor[5]. This is the second layer of our deployment. Each bridge is responsible for receiving packets from a subset of the deployment's devices based on proximity. The collected packets are then transmitted, without any attempt to decrypt them, to the LoRaWAN where they are picked up by the LoRa gateways available in the area. The LoRa gateways together with the LoRa Server and the edge processing services form the 3rd layer of our setup, with devices based on the Raspberry Pi[6] single board computer. In this layer the received packets are decrypted and decoded based on the packet format defined by the meter's manufacturer. Above all that, the 4th layer consists of the cloud services that finally collect all the data from the whole infrastructure and provide APIs and interfaces for accessing the collected data.

Our edge analytics platform is split into two parts, the edge-1 and edge-2 levels. The edge-1 level is capable of communicating only with a limited number of devices, due to its low computation power (1-6 water meters). Its main job is to collect packets from the water meters, identify the source of each message and prioritize its upload to the higher layers of the system, as well as control of the remote controller valves.

On top of that, the edge-2 level possesses much more capable devices that can process and analyze a lot more data. The edge-2 processing services include:

1. A service for analyzing incoming packet rates and the signal quality from the installation's meters.
2. A key management service for storing and accessing the meter's decryption keys.
3. A service for producing analytics on received sensor data.
4. A local storage layer for storing the generated analytics.
5. A service for syncing data to the central cloud infrastructure.

In the rest of this paper, we focus on the operation of components 1 and 3 to showcase the real-time analysis of the incoming data packets from the water meters installed. The analysis of the data is done using Apache Flink[7] on the low cost Raspberry Pi single board computers. For analyzing the incoming packet rates, our goal is twofold. On the one hand, we want to find irregularities in the reported data (i.e., water consumption, flow and pressure) from each meter and on the other hand we try to fill in missing sensor data due to problematic communication between the water meter and the bridge devices of the installation. Based on the irregularities we find we can decide whether the data collected are going to be directly delivered to the cloud services of our system to produce any kind of alerts or notifications to the users and administrators of the system or they are going to be collected to be sent later on the day. For the analysis of sensor data in the cloud services we need to generate aggregated metrics on the water consumption and the water pressure reported on different time granularities. All operations are implemented using Apache Flink Stream Analysis.

4 Data

4.1 wM-Bus data packets

The data transmitted by the water meters and the rest of the devices in our deployment follow the specification defined by the wM-Bus protocol [1]. Each meter transmits periodically a single wireless frame that contains

[5] https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html
[6] https://www.raspberrypi.org/
[7] https://flink.apache.org/

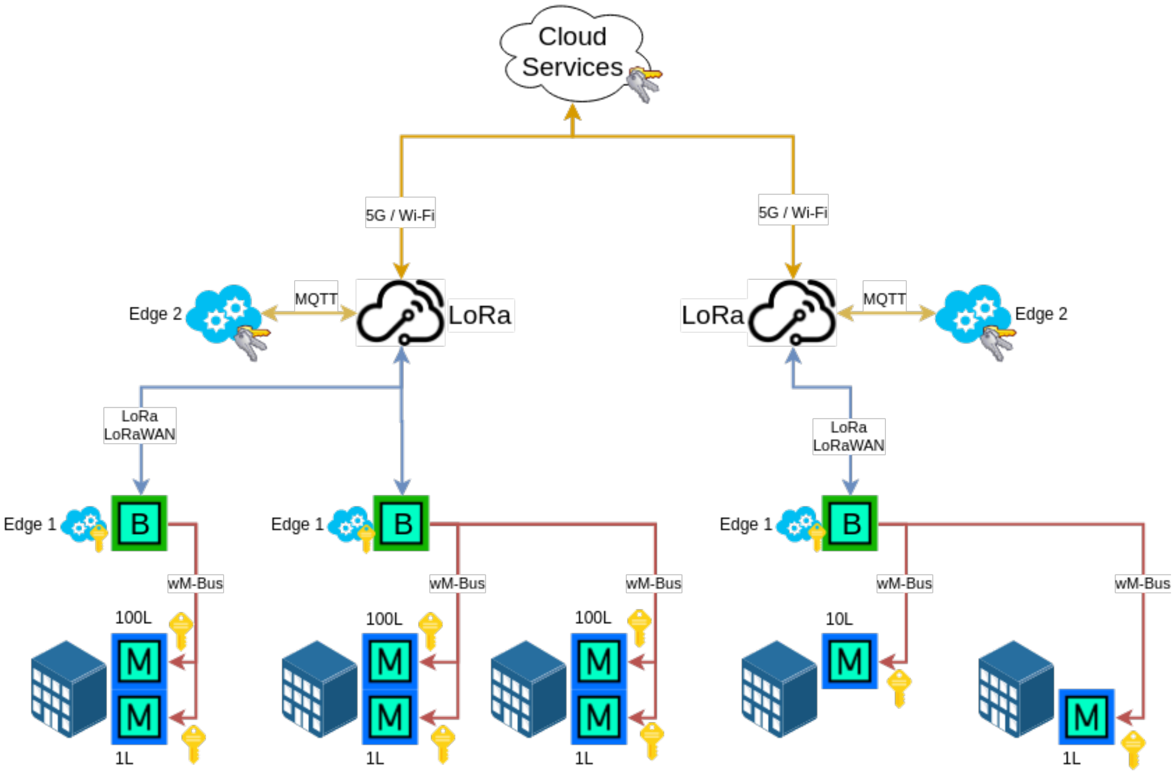

Figure 1: The smart meter data collection evaluation setup.

two parts. The header part is unencrypted containing information about the device's identifier and the format identifier of the encrypted data. The payload part is encrypted using a unique key for each device. Once decrypted, this part depending on the format defined in the header can contain information about the water volume measured, the water flow, the temperature of the water and the environment as well as alarms about the valve's status (e.g., whether someone has tried to tamper with it or physically tried to stop the measurement). Each meter transmits different packet formats at predefined intervals that range from 3 minutes to 1 hour, plus some randomized offset due to clock drift to avoid collisions. Packet transmission events from two water meters by two different manufacturers are presented in Fig. 2. As we can see the first meter presented on the left broadcasts messages much more frequently than the second one while the contents inside the packets follow the same format.

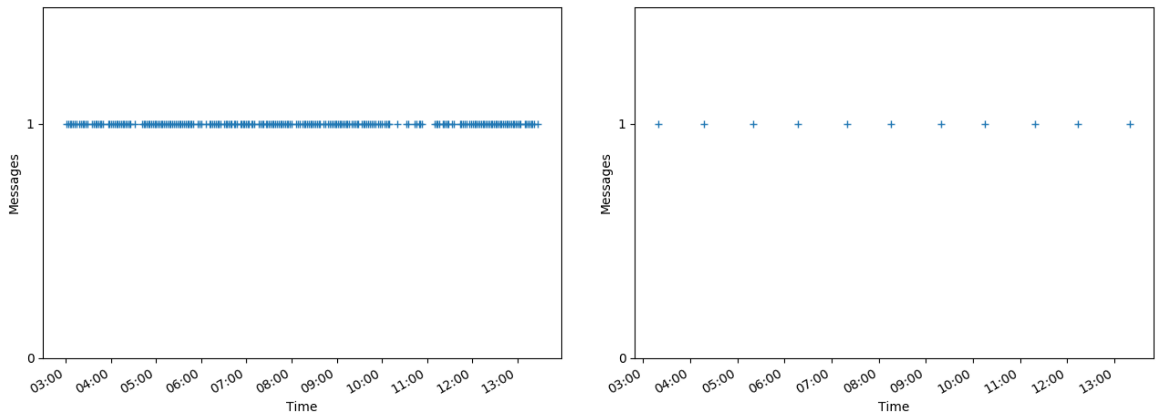

Figure 2: Packet transmission events for two different types of water meter devices for the same time period.

4.2 Packet Rate and Signal Quality Analysis

To generate analytics on the packet rates and signal quality, the system generates for each packet the time interval since the last one received and its received signal strength indicator. For these metrics, it then computes its average and standard deviation statistics. Using this, the system can detect whether the currently reported time interval (or signal quality) is normal or not based on the assumption that a value in the $[avg - 2*std, avg + 2*std]$ is considered acceptable. These abnormal values are called outliers and are dropped from any further processing while a notification is generated for the system administrator to indicate an unhealthy of the deployment. The data that are valid are used to adapt the calculated average and standard deviation values to incorporate cases where the average time interval of the packet reception changes over time. Outliers on the signal quality indicate that a device while operating and transmitting data for the moment could be facing a problem in the future as its signal is degrading due to environmental reasons or any external interference.

4.3 Sensor and Meter Data Analysis

The analysis of the sensor and meter data is the target of the whole operation of our system. In this case, we do not need to exclude data from further analysis since the data reported from the meters are trusted as accurate but we need to generate alarms for our end users if the received data deviate from the expected behaviour of the meter (consumer). For example, detecting an abnormal water flow, much higher than normal, could indicate a broken pipe that needs to be fixed and could incur unexpected charges on the final client.

5 Scaling up to a Smart City

To conduct our evaluation we used the data from our real-world deployment and scaled it up to reach the conditions to be faced in fully fledged Intelligent City installations of different sizes. The deployment would be much more dense and the total generated data could exceed the processing capabilities of a single cloud infrastructure. To get estimates on how many devices could actually be deployed in a real world city we follow the categorization provided in [6] and actual data from water distribution networks [8]. Based on that data, we can categorize cities in 5 different categories based on their population (Small to XXLarge) presented in Table. 1.

City Category	Population Limits	Water Meters
Test Site	-	48
Small	between 50000 and 100000	between 27000 and 54000
Medium	between 100000 and 250000	between 54000 and 135000
Large	between 250000 and 500000	between 135000 and 270000
XLarge	between 500000 and 1000000	between 270000 and 540000
XXLarge	between 1000000 and 5000000	between 540000 and 2700000

Table 1: City categories to be used in our evaluation.

In our evaluation setup a total of 3000 packets are received by all the deployed meters every day. These packets generate multiple measurements but for the rest of our evaluation we will keep referring to the number of packets instead of the sensor measurements for simplicity. To scale this number from our evaluation setup to the city categories we start to see the benefit of using such a distributed processing infrastructure. The expected packets per day and data sizes are available in Table 2. We use the lower estimates for the number of deployed water meters in each city category to calculate the number of expected packets and data sizes. As we can see even from a small city with a population of 50000 and 27000 water meters to collected data exceed 13GB, a load big enough for any network or processing infrastructure.

5.1 PreProcessing data on the edge

Pre-processing each message directly on the cloud requires establishing or maintaining a communication channel with constant data flow from the edge devices of our system to the remote cloud infrastructures used. Such a connection is not the best option especially when communication is done over metered connections (e.g., a 5G network).

On the contrary, we chose to do the data pre-processing on the edge devices already available in the installation (Raspberry Pis running the LoRa server software). Running the analysis on the same amount of data on the

[8] https://www.eydap.gr/en/TheCompany/Water/DistributionNetwork

Location	Packets per Day	Data Size	Gzip	Bzip2
Test Site	3000	24 MB	6.3 MB	5.3 MB
Small	1687500	13.5GB	3.5GB	2.9GB
Medium	3375000	27GB	7GB	5.9GB
Large	8437500	67.5GB	17.7GB	14.9GB
XLarge	16875000	135GB	35.4GB	29.8GB
XXLarge	33750000	270GB	70.8GB	59.6GB

Table 2: Size of collected data for a single day from the evaluation deployment and estimates for city-wide installations and compression benefits.

edge takes significantly more time, around 30 seconds (vs 3.5 seconds in the cloud server) but saves a lot of the generated traffic. Using this method, we can combine multiple packets over larger time intervals and transfer them, all together in a compressed format to the cloud. Due to the repetitive nature of the collected data, compressing them could result in huge gains over the final size of the data that needs to be uploaded. Also, due to our edge pre-processing, we can identify situations when there is a need for urgent communication and trigger an immediate upload of the data collected so far. As seen in Table 2 the size of data that needs to be uploaded to the cloud every day reaches a total of 24 MB. Compressing the data could save up to 80% on data to be uploaded in total, every day when no immediate uploads are required leading to much more important gains in larger scenaria.

6 Conclusions

This paper presented the properties of a real world smart metering solution combined with an edge processing and analytics solution for collecting and analyzing the data produced in the edge of the network. Our solution uses the intermediate layer between the IoT deployment and the cloud services deployed in large datacenters to alleviate a series of issues in the areas of scalability, bandwidth consumption reduction while providing seamless operation for the whole system. Then based on the data from real world water metering networks, we estimate the amount of data a fully fledged smart city solution will need to handle, showing how our solution fits in the bigger picture.

References

[1] Communication systems for meters. https://ec.europa.eu/eip/ageing/standards/ict-and-communication/data/en-13757_en, accessed: 2019-09-06

[2] Akrivopoulos, O., Amaxilatis, D., Chatzigiannakis, I., Tselios, C.: Enabling stream processing for people-centric iot based on the fog computing paradigm. In: 2017 IEEE 22nd International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–8 (Sept 2017). https://doi.org/10.1109/ETFA.2017.xx

[3] Akrivopoulos, O., Chatzigiannakis, I., Tselios, C., Antoniou, A.: On the deployment of healthcare applications over fog computing infrastructure. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). vol. 2, pp. 288–293 (July 2017). https://doi.org/10.1109/COMPSAC.2017.178

[4] Bottaccioli, L., Aliberti, A., Ugliotti, F., Patti, E., Osello, A., Macii, E., Acquaviva, A.: Building energy modelling and monitoring by integration of iot devices and building information models. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). vol. 1, pp. 914–922 (July 2017). https://doi.org/10.1109/COMPSAC.2017.75

[5] Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (Jan 2008). https://doi.org/10.1145/1327452.1327492, `http://doi.acm.org/10.1145/1327452.1327492`

[6] Dijkstra, L., Poelman, H.: Cities in europe: the new oecd-ec definition. Regional Focus **1**(2012), 1–13 (2012)

[7] Fasolo, E., Rossi, M., Widmer, J., Zorzi, M.: In-network aggregation techniques for wireless sensor networks: A survey. Wireless Commun. **14**(2), 70–87 (Apr 2007). https://doi.org/10.1109/MWC.2007.358967, `http://dx.doi.org/10.1109/MWC.2007.358967`

[8] Hirzel, M., Soulé, R., Schneider, S., Gedik, B., Grimm, R.: A catalog of stream processing optimizations. ACM Comput. Surv. **46**(4), 46:1–46:34 (Mar 2014). https://doi.org/10.1145/2528412, `http://doi.acm.org/10.1145/2528412`

[9] Hlaing, W., Thepphaeng, S., Nontaboot, V., Tangsunantham, N., Sangsuwan, T., Pira, C.: Implementation of wifi-based single phase smart meter for internet of things (iot). In: 2017 International Electrical Engineering Congress (iEECON). pp. 1–4 (March 2017). https://doi.org/10.1109/IEECON.2017.8075793

[10] Karthikeyan, S., Bhuvaneswari, P.T.V.: Iot based real-time residential energy meter monitoring system. In: 2017 Trends in Industrial Measurement and Automation (TIMA). pp. 1–5 (Jan 2017). https://doi.org/10.1109/TIMA.2017.8064790

[11] Morello, R., Capua, C.D., Fulco, G., Mukhopadhyay, S.C.: A smart power meter to monitor energy flow in smart grids: The role of advanced sensing and iot in the electric grid of the future. IEEE Sensors Journal **PP**(99), 1–1 (2017). https://doi.org/10.1109/JSEN.2017.2760014

[12] Papageorgiou, A., Poormohammady, E., Cheng, B.: Edge-computing-aware deployment of stream processing tasks based on topology-external information: Model, algorithms, and a storm-based prototype. In: 2016 IEEE International Congress on Big Data, San Francisco, CA, USA, June 27 - July 2, 2016. pp. 259–266 (2016). https://doi.org/10.1109/BigDataCongress.2016.40, `http://dx.doi.org/10.1109/BigDataCongress.2016.40`

[13] Tselios, C., Tsolis, G.: On QoE-awareness through Virtualized Probes in 5G Networks. In: Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on. pp. 1–5 (2016)

[14] Xu, G., Ngai, E.C.H., Liu, J.: Ubiquitous transmission of multimedia sensor data in internet-of-things. IEEE Internet of Things Journal **PP**(99), 1–1 (2017). https://doi.org/10.1109/JIOT.2017.2762731