future internet

Article

A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access

Pape Abdoulaye Barro [1,*], Marco Zennaro [2], Jules Degila [1] and Ermanno Pietrosemoli [2]

[1] Department of Information and Communications Technology, Institute of Mathematics and Physical Sciences, Porto-Novo 3730-301, Benin; jules.degila@imsp-uac.org

[2] T/ICT4D Laboratory of the Abdus Salam International Centre for Theoretical Physics, 34151 Trieste, Italy; mzennaro@ictp.it (M.Z.); epietros@ictp.it (E.P.)

[*] Correspondence: pape.barro@imsp-uac.org

Received: 20 February 2019; Accepted: 1 April 2019; Published: 8 April 2019

Abstract: An increasing number of implementations of IoT for development use the LoRaWAN protocol as many of them leverage the free network and application servers provided by The Things Networks (TTN) to fulfill their needs. Unfortunately, in some countries in Sub-Saharan Africa and South Asia, Internet access cannot be taken for granted, therefore, TTN might not be available. Moreover, low-cost and low-power consumption options devices are the most sustainable ones. In this paper, we propose a LoRaWAN network with autonomous base stations that can work without Internet connectivity for essential services, while being able to provide additional features whenever Internet access becomes available, even in an intermittent fashion. Security and privacy are preserved, with support for mobile nodes.

Keywords: community networks; ICT4D; IoT; LoRaWAN; Edge/Fog computing; wireless networks; IoT4D; smart city

1. Introduction

While IoT devices are popular in industrialized countries, their deployment in some developing countries faces additional hurdles such as the prohibitive cost of Internet access or its unavailability. The most appealing applications focus on environmental data: water and air quality monitoring, landslide detection, water levels in rivers, meteorological data and so on. In many countries, there is little information about the environment, creating a knowledge divide that adds to the digital divide. Those data can nevertheless be made available with a local infrastructure that can generate and publish the relevant information. Also, it is common to see people owning smart phones and tablets with no access to data, or not able to afford the extra cost charged by operators for data services. We then propose a sort of "*Intranet of Things*" in which data are locally generated and locally consumed even in the absence of Internet connectivity, or when Internet connectivity is either intermittent or painfully slow. Environmental data collected in the region by low cost and low power end-devices will be transmitted to one or more "Base Stations" using the LoRaWAN protocol [1] and made accessible through a WiFi access point or "Hot Spot" included in the Base Station (BS). Users can then have access to real time environmental data and other relevant information using their standard devices. They can also have access to stored static materials on health, education, legislation, environment protection and any other sort of content deemed of interest, including folklore and traditional knowledge shared by the community members.

Optionally, if there is Internet connectivity at the Base Stations, selected time-sensitive content can also be accessed by the whole community at a lower cost, given that a single Base Station can provide services to many users, performing then a "proxy" service. So, the Base Station will act as a LoRaWAN Gateway, a LoRaWAN Network Server, a LoRaWAN Applications Server, a local data and content repository, a Bulletin Board and a WiFi Access Point that can be accessed by any WiFi client device. Since the Base Station cannot be sleeping at any time, its power consumption is considerably greater than that of the end-nodes, but it is nevertheless moderate and easily accommodated by means of a lightweight photovoltaic electric power system (EPS). Despite its many functions, it is also cheap since it is based on a Raspberry Pi with a LoRaWAN hat and Open Source software. Optionally, the BS can be connected to Internet through an appropriate router, in which case, the connection costs could be shared among its several users.

1.1. LoRa

"LoRa Technology has revolutionized IoT by enabling data communication over a long range while using very little power. When connected to a non-cellular LoRaWAN^{TM} network, LoRa devices accommodate a vast range of IoT applications by transmitting packets with important information. LoRaWAN fills the technology gap of cellular and Wi-Fi/BLE based networks that require either high bandwidth or high power, or have a limited range or inability to penetrate deep indoor environments. In fact, LoRa Technology is flexible for rural or indoor use cases in smart cities, smart homes and buildings, smart agriculture, smart metering, and smart supply chain and logistics" [2] (see Figure 1).

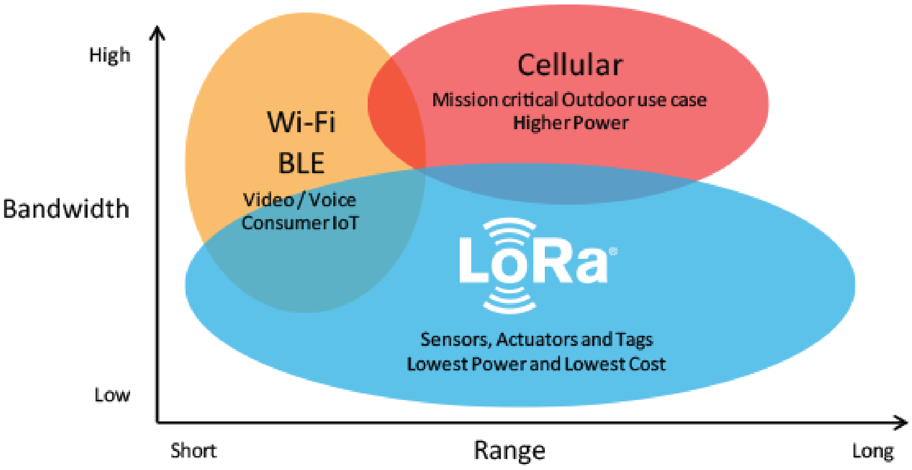
High
Bandwidth
Low
Wi-Fi
BLE
Video/Voice
Consumer IoT
Cellular
Mission critical Outdoor use case
Higher Power
LoRa
Sensors, Actuators and Tags
Lowest Power and Lowest Cost
Short
Range
Long

Figure 1. Why LoRa? LoRa has a greater power budget than Cellular or WiFi and therefore can achieve a greater range or deeper in building coverage, by using a much lower data rate.

1.2. LoRaWAN

Many IoT applications, especially in rural regions, need to span considerable range while consuming as little power as possible. LoRa modulation can achieve these goals by spreading the transmission spectrum to, say 125 kHz, but limiting the throughput to a few kilobits per second at most. The relationship between these two variables is called the spreading factor (SF) and it can be varied to suit different needs. A low SF allows higher throughput while a high SF allows decoding signals even below the receiver's noise floor, thus permitting ranges of dozens of kilometers in unencumbered environments and a few kilometers even in urban areas. LoRaWAN is an open media access control (MAC) layer protocol that provides all the functionalities for a complete solution, based on multiple end-nodes that communicate to one or more LoRaWAN Gateways. Gateways in turn connect to a Network Server (NS) using any kind of IP protocol based links. For instance the Gateway might have a cellular modem and SIM card, to connect to the Network server, or it might use a WiFi transceiver and a high gain antenna, or have an Ethernet port to connect to a local wired infrastructure. It can also use a satellite link, by paying the corresponding connection fees. The NS also connects to the

Application Servers (APs) using the IP protocol over whatever infrastructure is available as shown in Figure 2. Gateways are transparent bridges relaying messages between end-devices and one or more network servers. Payload is encrypted by the end-node using the 128-bit Advanced Encryption Standard (AES), so each application is isolated from the others. Traffic is bidirectional, but mostly upward, from the end-node to the Gateways with very little in the downlink direction. A significant advantage of LoRaWAN is the fact that it uses unlicensed bands and open protocols so that it can be implemented by any organization independently from any commercial service providers.

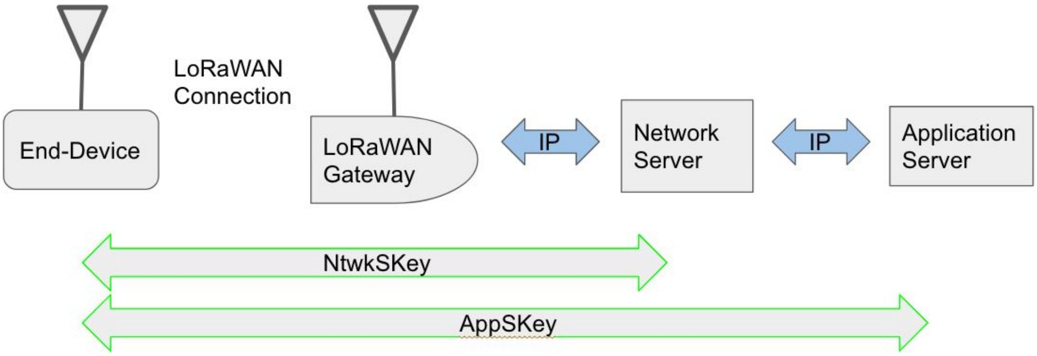

Figure 2. LoRaWAN Secure Architecture.

The LoRaWAN specification defines three device types [3]. All LoRaWAN devices must implement Class A, whereas Class B and Class C are extensions to the specification of Class A devices. A class A device will only listen for possible downlink messages from the Gateway after completing the transmission of its up link messages, and only for two briefs time windows. This will allow the device to sleep most of the time, therefore achieving the maximum battery duration. A class B device will also wake up at certain specified intervals, so its battery will last less. Class C devices will be listening whenever they are not transmitting, so their consumption is the highest, and they are normally meant to be attached to the power grid or a photovoltaic system. All devices and applications have a 64 bit unique identifier (DevEUI and AppEUI). When a device joins the network, it receives a dynamic (non-unique) 32-bit address (DevAddr). The NwkSKey is used to validate the integrity of each message by its Message Integrity Code (MIC) and the AppSKey is used for encryption of the application payload.

1.3. TTN—The Things Network

Part of the success of LoRaWAN is due to the growth of a public community network known as The Things Network (TTN) [4]. TTN enables low power end-nodes to leverage existing gateways to connect to an open-source network which enables access to the user's applications. It is a decentralized and collaborative network, counting thousands of gateways around the world, used by tens of thousands of developers and businesses to build valuable use cases. Developers can use existing gateways, or add new ones where additional coverage is needed. Gateways have to be connected to the Internet to be able to access the TTN server that runs the required MAC functions. There are many servers worldwide, based on several cloud technology platforms. While there are over 6000 gateways up and running worldwide, there are less than 100 in Africa, hindering access to this open and free network where it is mostly needed.

1.4. Connectivity

Connectivity is still an issue in many developing countries. Nearly nine out of ten young individuals with no access to Internet live in Africa, Asia or the Pacific according to the latest ITU statistics [5]. Connecting in a stable way to a cloud service such as the one offered by TTN is not feasible in many developing countries rural or isolated areas.

The collected data at https://www.ripe.net on the round trip time (RTT) for packets originating from some African countries to the Ripe servers in Europe and back to the countries, show significant high Internet access latencies, which hinders many real time applications (see Figure 3).

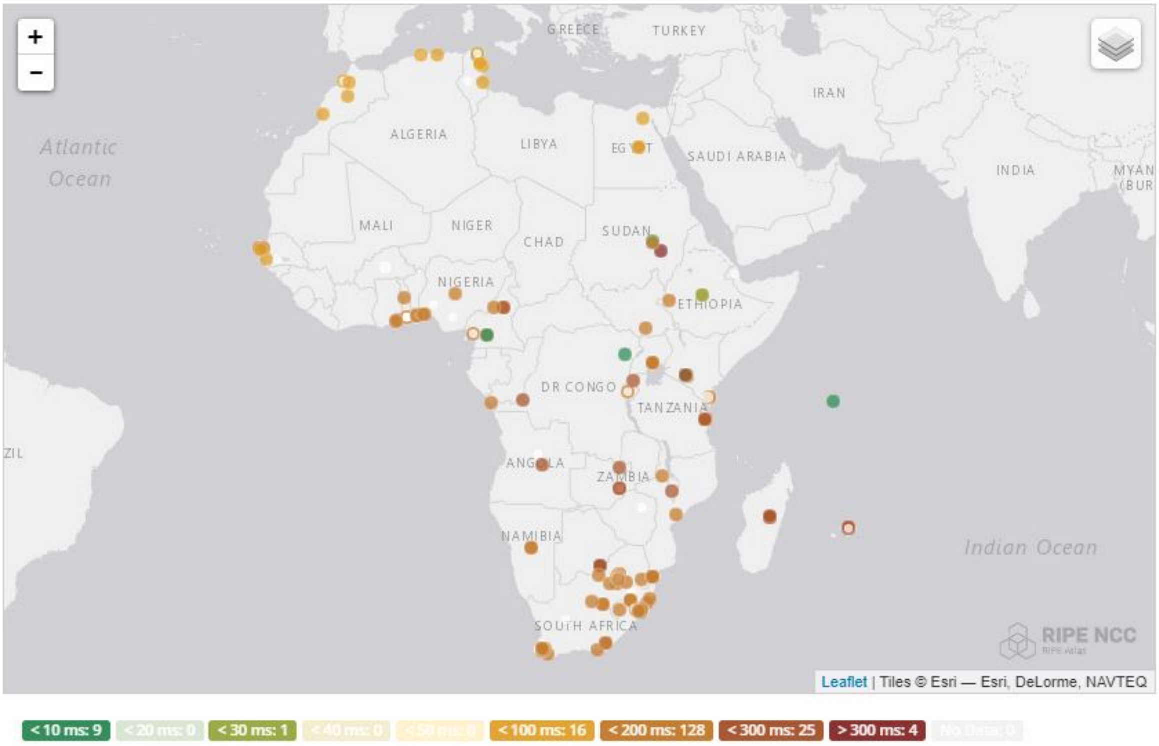

Figure 3. Internet latency in Africa.

In Figure 3, very few green dots indicates quick connection to Internet and most of the dots range from light brown to dark brown showing higher latency connections from most African countries. The analysis performed on the Ripe data also reveals that the round trip time can exceed 6 seconds, which is not tolerable for real-time applications.

2. Scenarios

To illustrate the typical requirements, we use the following scenarios with some potential users such as Massamba, a citizen concerned about the air quality around him, little Mademba, the schoolboy that needs access to educational materials, Yao, the businessman running a taxi company, grandfather Tossa who requires continuous health monitoring device and the Benin Government that has installed distributed base stations in the city of Cotonou to sense the air quality as well as to gather the electrical consumption measurements. Yao needs to know the position of his taxis even where there is no Internet access. Grandfather Tossa has a heart disease and is fitted with a heart rate monitoring device that can send alert when urgent assistance is required.

These examples show the diverse needs that can all be met with a proper data communication network. In Section 3, we describe some of the previous related works. In Section 4, we deal with the requirements of the proposed Base Stations, while its general architecture is described in Section 5, with the proof of concept in Section 6. Performance tests are described in Section 7, and conclusions are in Section 8.

3. State-of-the-Art

The main purpose of this research is to develop autonomous base stations leveraging the LoRaWAN protocol for a wide range network, minimizing deployment costs related to Internet usage and also allow isolated communities to get some services even without having Internet access. This work was initially motivated by the desire to offer to citizens of developing countries the possibility of being able to measure the quality of the air and, thus, urge their government agencies to undertake appropriate policies and actions to ensure a healthy and sustainable environment. The review of major relevant publications showed that all of them take Internet connectivity for granted. This is the case in:

- Reference [6] which proposes a system to monitor external environmental factors such as temperature, humidity, and some gases using a Raspberry Pi as base station and the ESP8266 Wi-Fi Module (at the end devices level) to collect data and storing them on MySQL servers over the Internet. Since the WiFi coverage of the base station is not wide, this solution is not amenable for large-scale coverage.
- Reference [7] which proposes an air quality monitoring system in the city of Dhaka (Bangladesh), using mq-2, mq-3 and mq-7 sensors attached to a micro-controller which is connected to the Internet. [6] seems to be more adoptable than this one.
- Reference [8] which presents a participatory urban detection framework for monitoring PM2.5 (Particulate matters up to 2.5 mm). Their framework is based on open hardware, open source software and open data. They were able to develop four Internet-connected PM2.5 monitoring devices with different design considerations based on previous results. The approach adopted here is original, the only drawback is the use of the Internet at every collection points and therefore not optimal in terms of cost for wide deployment and not feasible for isolated communities.
- Reference [9] which developed a system for assessing air quality using the BeagleBone equipped with the MQ-7 sensor to measure CO, MQ-11 to measure H2, and a GPS to record position. With this system, if the Internet is interrupted, access to the data is lost.
- Reference [10] which measures environmental parameters such as temperature, humidity, light intensity and air quality using a Wio-Link board and Raspberry Pi (access to the Internet), then publishes this data to an MQTT broker. We can see the same difficulties as in [6], except that here their solution is oriented notifications by email or sms.
- Reference [11] which presents an autonomous real-time air quality monitoring system that registers PM2.5, carbon monoxide, carbon dioxide, temperature, humidity and atmospheric pressure. They use an Arduino uno connected to a Raspberry Pi (by USB) that routes data to the cloud via the Internet. We can see the same limitations here. Furthermore, authors claim that their solution is real-time. This challenges its usefulness in many African countries due latency issues.
- Reference [12] which leverages the MQ7 and MQ135 sensors, the ESP8266 WiFi module and the Nucleo F401RE microcontroller to measure gases such as CO, CO2, SO2 and NO. An NoSql Mongo database is housed in a Raspberry Pi 3 that functions as a gateway for the WiFi connected nodes. The base station also functions as a web server for data visualization. This solution is usable both locally and via the Internet, but the coverage of the base stations is still limited, thus, limiting its application in a significant range.

In all these references, Internet access is assumed at most data collection points. This cannot be taken for granted in isolated communities or in places with high Internet latency. This suggests that the word "Internet" in "Internet-of-Things" should not be taken literally. In developing countries, Internet is still a scarce resource and even where it is available, it might not be affordable for a majority of the population [13]. The challenge is, then, to make "Internet-of-Things" accessible to the underprivileged [14] by leveraging the LoRaWAN protocol.

The concept that allows extending the coverage or the feasibility of a city-wide coverage network based on autonomous LoRaWAN Base Stations in which only the Master Base Station needs to be connected to the Internet is what distinguishes our proposal. Most of the LoRaWAN gateways encountered on the market are designed to reach TTN or other servers by means of the Internet. An open-source LoRaWAN Gateway described in [15], collects the LoRa packets from the end-nodes and transports them to the network server over IP using the MQTT protocol. This research is therefore a first step towards building sustainable infrastructure to extend coverage to places with limited connectivity like it's the case in many regions of Africa.

Charith [16] states that "in reality sensing all possible data items captured by a smart object and then sending the complete captured data to the cloud is less useful. Further, such an approach would also lead to resource wastage (e.g., network, storage, etc.)". He also states that "The Fog (Edge) computing paradigm has been proposed to address this weakness by pushing to the edges processes of knowledge discovery using data analytic. However, edge devices have limited computational capabilities". "We agree with him, and we also believe that an autonomous base station, despite its limited resources, must be able to first process the data locally, make it available to the users in real time, before possibly forwarding it to the core network for wide scale management". This study is therefore oriented in this direction and will allow the implementation of the minimal and cheaper infrastructure for the feasibility of the smart city in developing countries. This infrastructure must allow citizens to install their end nodes and then access their own information. Data can also be shareable. We propose, then, a multi-services edge gateway for communities with three variants: a standard box always connected to the Internet, a box connected intermittently over a delay tolerant network and a standalone box catering to the essential needs of isolated networks. In all these three cases, community members can use their smartphones or tablets to access the data of their interest, as well as additional content, through the WiFi Access Point embedded in the Base Station.

4. Requirements of the Proposed BS

1. *low cost*: As computing platform we selected the Raspberry Pi single board linux device which is inexpensive and widely available worldwide. The LoRa capabilities are provided by two hardware additions: An eight channel module from ISMT that costs 200 euro or a single channel module from Uputronics that is only 26 euro. The WiFi radio is built-in, and the Ethernet port can be used for internet access or for connection to a local network.
2. *low power*: The Gateway must be always on to receive messages from any end-node, so it must be power frugal, specially in the receiving mode, which will be the predominant one. Since even where grid power is present we anticipate the probability of outages, we suggest the use of a rechargeable battery, supplemented by a photovoltaic panel where grid power is unavailable or intermittent.

   The end-nodes will be sleeping most of the time, so their battery will last very long, the exact duration depending on the consumption of the attached sensors and the frequency of transmissions.
3. *expandable storage*: Swapping the SD card for a higher capacity one allows the storage of extra data and content.
4. *open source*: Only open source software is used, so other researchers/organizations can replicate the design and expand it to meet specific needs.
5. *Unlicensed bands*: Communications is done on ISM bands, usable worldwide free of charge.

5. Architecture

The above requirements led us to the architecture shown in Figure 4.

The architecture encompasses two access modes: a local one (comprising the measurement layer, the link layer and the InterTrans interfaces) and a remote one (from the InterTrans interface to the application layer through the middleware and the backup layers). Between the lower part of the

RadioCom interface and the MonitorServ interface, we have the core network which includes the link, middleware, and backup layers.

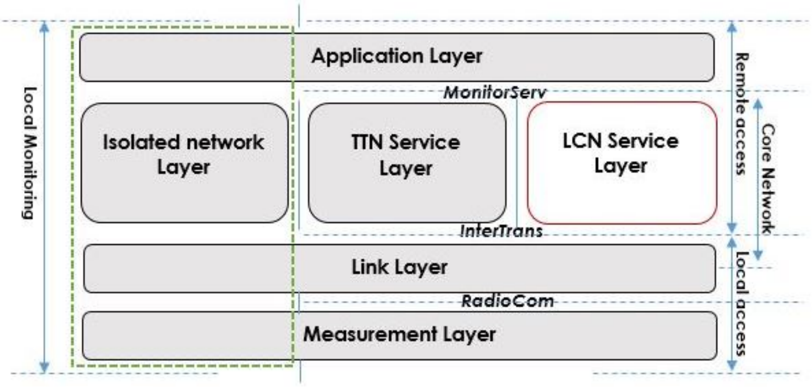

Figure 4. General architecture.

On the left of Figure 4 we have the isolated network layer in which the microcomputer requests services from the link layer to accommodate demands from the application layer via its MonitorServ interface. The Link Layer also provides access to the TTN service layer and/or network layer of the local community. The measurement layer provides data collection services to the link layer (see Figure 5).

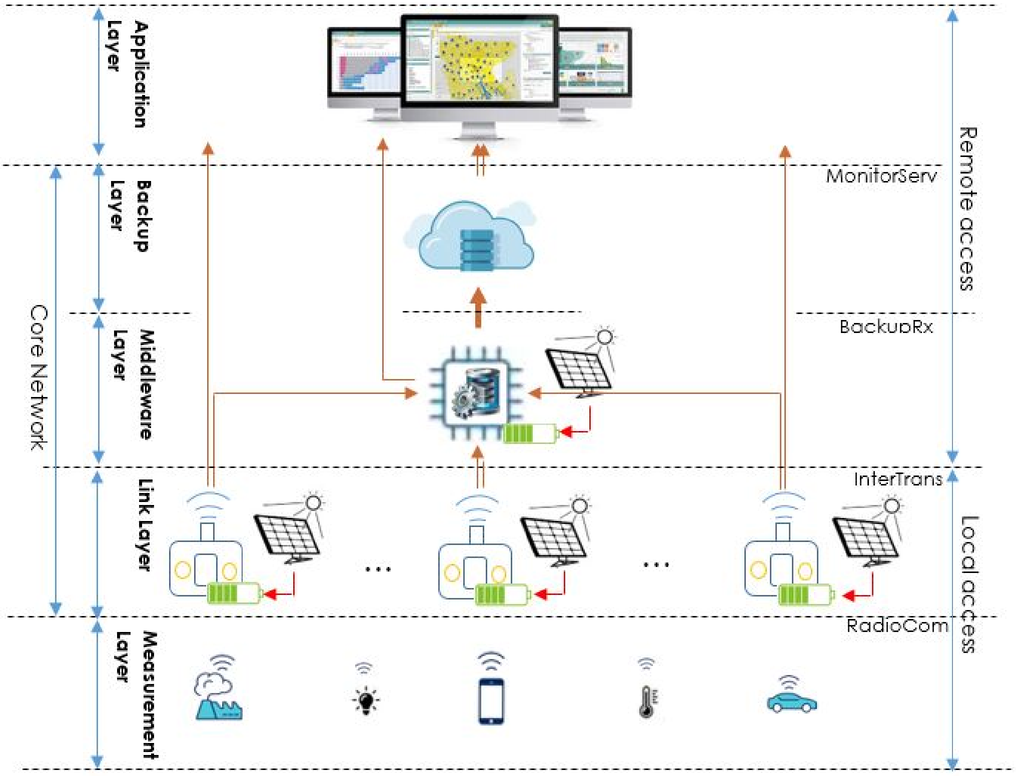

Figure 5. Detailed architecture.

6. Proof of Concept

When several Base Stations which can communicate among themselves but do not have Internet access are deployed, the problem of data consistency arises. This can be solved by designating a Master Base Station to which all the other stations (now called slaves) must refer to disseminate any change of content. This will also support roaming of a mobile node as shown in Figure 6.

The Master acts as an authoritative data base and, through its LoRa-gateway bridge, centralizes the data coming from the slave base stations. Any change in a Slave station will reach the Master base station, which also provides coverage of its surrounding area.

The Slaves can choose to be their own Master or can join an existing one. They thus have the capability of managing their own data but can also contribute to a community network (LCN). A community can choose its coverage area and that of the roaming mobile nodes.

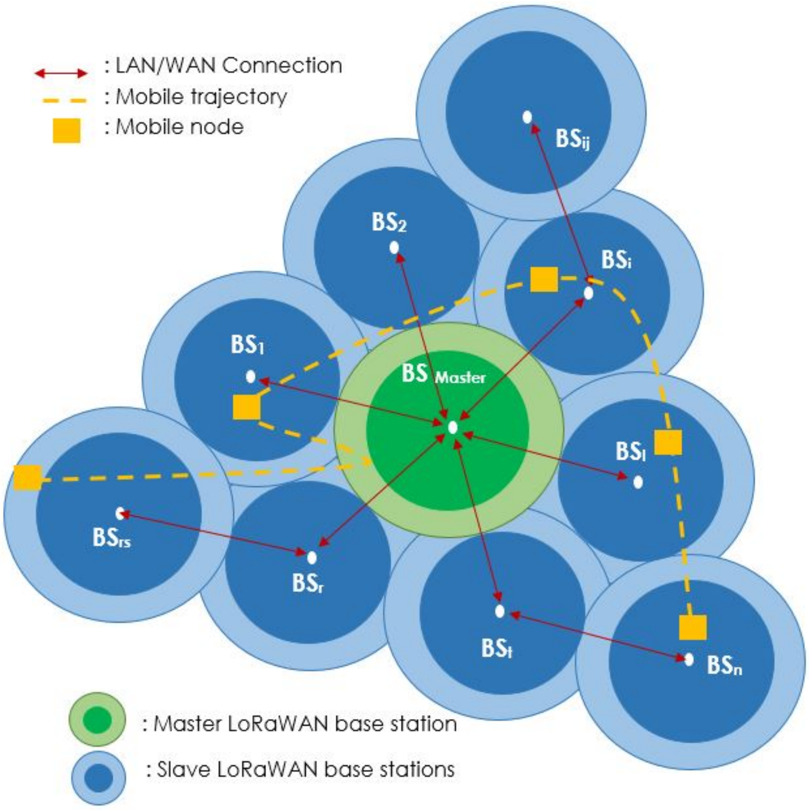

Figure 6. Operating principle.

The architecture of Both Master and Slave base stations comprises the four parts shown in Figure 7: LoRaWAN Gateway (GW), Network and Application servers, TIG (Telegraph, Grafana, Influx) and WiFi Access Point with content server.

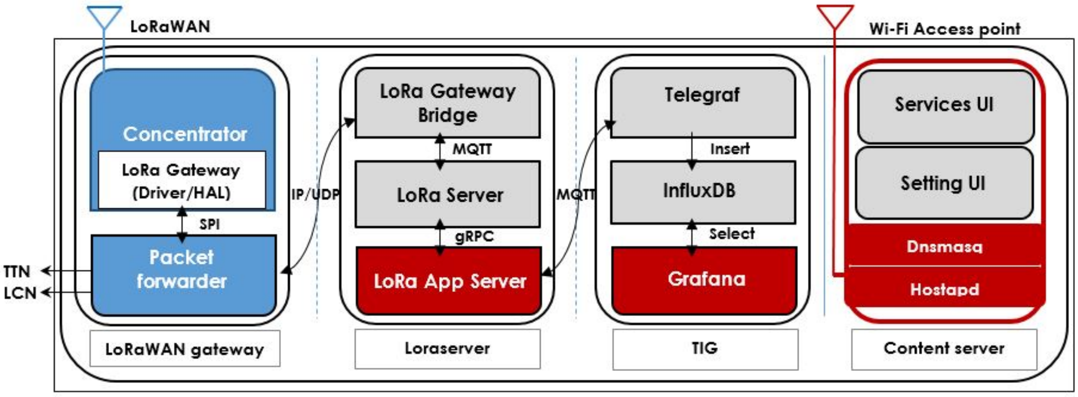

Figure 7. Four main parts of the Base Station.

LoRaWAN is used to collect data from the end nodes and WiFi to communicate final users to the web server through the Access Point. The 8 channel Base Station is shown on the left of Figure 8 and the end-node on the right. In the middle is the single channel Base Station which constitutes the cheaper and less power consuming solution.

1. The *LoRaWAN gateway* acts as packet forwarder between the end nodes and the Network server, based on the open implementation proposed by Semtech, and built with a Raspberry Pi and additional hats. For the eight channel version, the concentrator hat is based on the SX1301 chip which allows listening simultaneously in eight different frequencies with several orthogonal spreading factors. The single channel gateway uses a cheaper hat based on the RFM96W RF module (see Figure 8).
2. *Loraserver*: The role of the network server *(LoRa Server)* is to authenticate nodes to make sure that only data from registered nodes will be forwarded. It is also responsible for the deduplication and processing of uplink frames, MAC layer processing and downlink transmission planning. The application server *(LoRa App Server)* provides payload processing and decryption. It offers to the end user the requested data, like weather, air quality index, alarms and so on. Data are locally processed, stored, and made available through the WiFi Access Point.
3. *TIG*: The Telegraf tool is used to subscribe to the MQTT Mosquitto broker, to collect the messages (in JSON format) transmitted by the LoRa App Server and to inject it into the InfluxDB time series database. By using the retention strategy proposed by InfluxDB, the data of a given period can be automatically deleted to save storage space. This can also be done manually on the Service UI. Using Grafana open platform, measurements can be retrieved on influx DB for analysis and monitoring.
4. The *Content Server* provides access to both static content like educational material, health information, farming best practices, etc., as well as Community Bulletin Board services. The WiFi Access Point makes this material available to the community through a web interface.

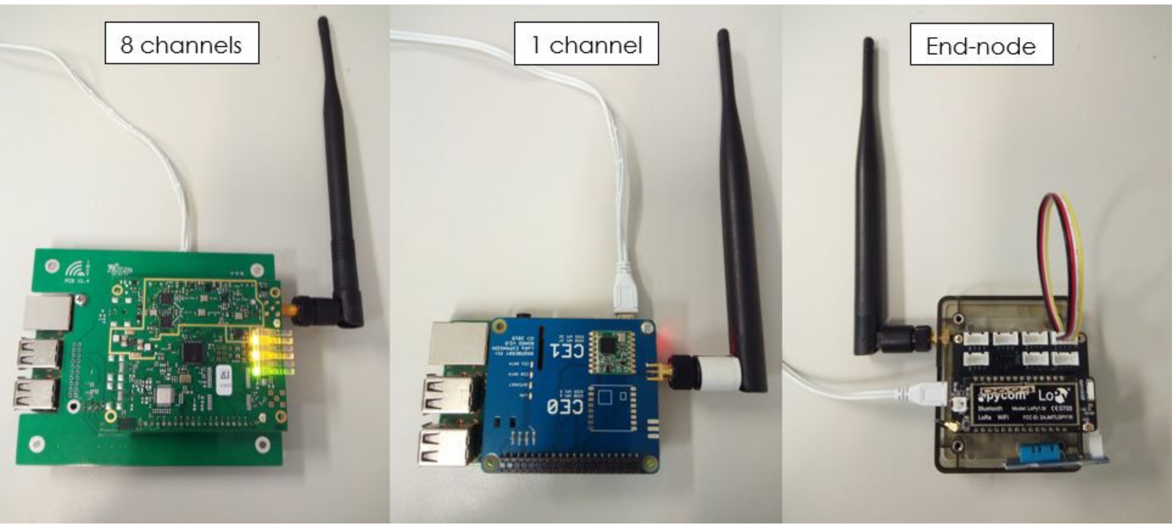

Figure 8. BS with eight channels gateway (**left**), BS with single channel GW (**middle**) and end-node (**right**).

The output of the eight channels concentrator is sent to the Packet forwarder (PF) via a Serial Peripheral Interface (SPI). The PF transfers it to the LoRa Gateway Bridge by IP/UDP (see Figure 9). We wrote the edge-gateway.service that invokes a start.sh script which installs and executes the dependencies specified in the Mainfile and then launches the packet forwarder if all executions are successful. It also allows to upgrade the firmware as needed. For the single channel GW, everything is included in the packet forwarder, including the HAL library.

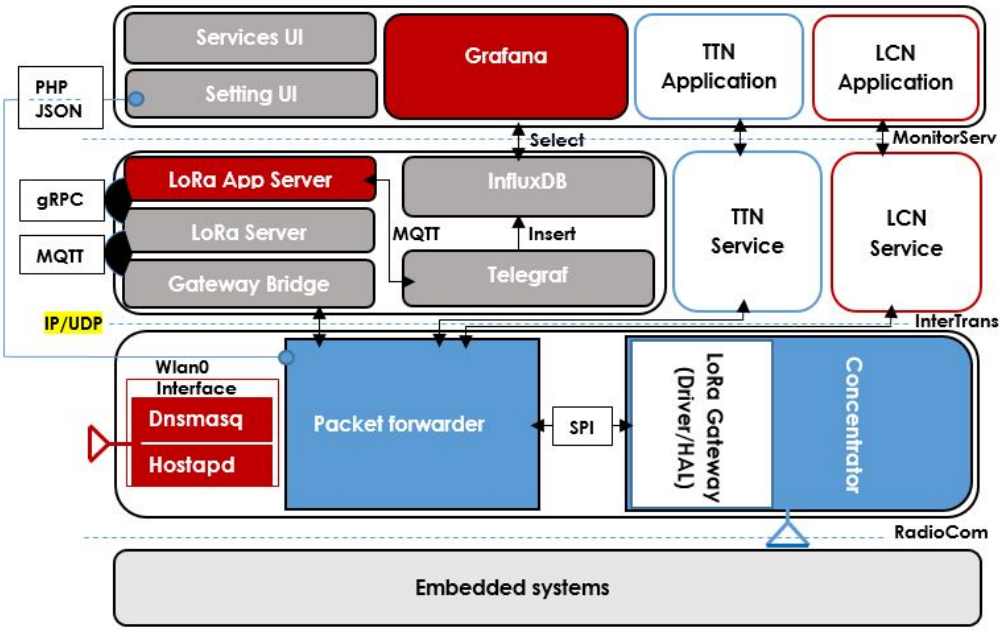

Figure 9. Base Station (BS) architectural choice.

Tools such as hostapd (WiFi Captive Portal) and dnsmasq (DHCP and DNS server) allow access to the graphical interface of the LoRa App Server (http://IP:8080) to register a node and grafana (http://IP:3000) to see the evolution of the measurements. Community content and Bulletin Board services are accessed through (http://IP).

The Raspberry Pi has a built-in WiFi interface configured as an Access Point and an SD card which is the local storage for data and content. The hats mentioned provide the LoRa functionalities for the complete Base Station.

Reference [17] covers proof of concept and additional details.

7. BS Performance Tests

For the tests, a LoPy LoRaWAN transceiver is chosen and fitted with a DTH11 temperature and humidity sensor. The Base Station (BS) uses a Raspberry Pi embedded computer with a LoRa hat, communicating on the 868 MHz unlicensed band with a bandwidth of 125 kHz. The power consumption and the CPU load of the BS were measured at a constant 5.0 V supply. The 8 channels BS had a peak power consumption of 5.2 W and an average of 4.5 W, while the single channel one had the same peak of 5.2 W but an average of only 1.8 W. This power can be provided by a photovoltaic panel connected to a battery for continuous operation. A fully charged 324 Wh battery will provide 3 days (72 h) of autonomy for the 8 channels gateway, while for the single channel gateway a 130 Wh battery will suffice.

Figure 10 shows CPU usage by different processes like CPU.user, CPU.system (indicating the amount of time used by the kernel, which is responsible for low-level tasks, such as hardware interaction, memory allocation, communication between operating system, running device drivers and managing the file system), CPU.nice, CPU.idle (which shows the unused CPU time), CPU.iowait (which marks the wait time for input or output operations), the CPU.softirq (which indicates the time that the processor spends processing interrupts).

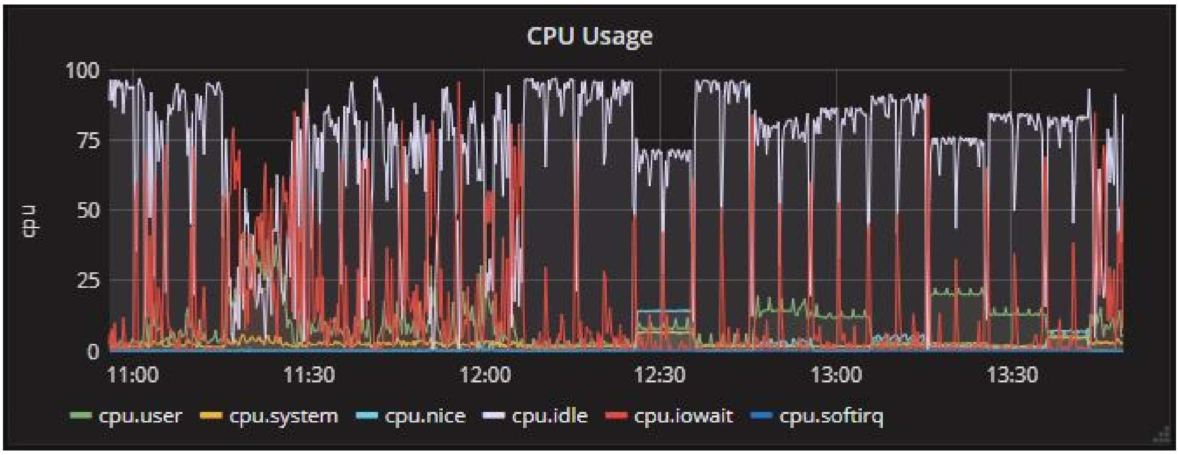

Figure 10. CPU usage by different processes.

8. Conclusions

This work aimed at extending the benefits of IoT to people living in areas with limited or no Internet connectivity, by collapsing the LoRaWAN servers normally spread in the cloud into a single box, the BS, that provides all the required functionalities. It can use TTN gateways and servers wherever they are available, even on an intermittent basis, or it can behave as a complete stand-alone system. Users interact with the proposed BS using their own devices over WiFi connectivity.

The end-devices are very cheap and battery powered, while the moderate power required by the always on Base Station can be supplied by an affordable photovoltaic power system. The single channel gateway is very inexpensive and power miser (therefore well suited for small communities), while the eight channels offers more capabilities at the expense of increased power consumption. The proposed storage for the prototype is a 16 GB SD card which can be expanded to add additional content. The inexpensive and widely available Raspberry Py is the core of the Base Station, and proved to be adequate for the required communication tasks. Two types of BSs were tested, a very inexpensive single channel device, that can handle moderate traffic, and a more expensive eight channels one that can be deployed when the traffic grows significantly. The BS slaves surrounding the BS master are interconnected via their packet forwarder and extend the community network allowing synchronization of content and also enabling roaming for mobile nodes.

A few of the many prospective IoT applications are sketched in the scenario's section. Future work will aim to address other applications for community networks that can leverage this kind of infrastructure to alleviate the existing disparity of opportunities in many countries such as those discussed here.

Author Contributions: Conceptualization, P.A.B., M.Z. and J.D.; methodology, P.A.B.; software, P.A.B.; validation, M.Z. and J.D.; formal analysis, P.A.B.; investigation, P.A.B., M.Z. and J.D.; writing–original draft preparation, P.A.B. and E.P.; writing–review and editing, M.Z. and J.D; visualization, J.D.

Funding: This research received no external funding.

Acknowledgments: We are grateful to the African Center of Excellence in Mathematical Science and Applications (CEA-SMA), IMSP, Dangbo, Republic of Benin, which supported this work, implemented at ICTP's Marconi Laboratory (ICT4D Lab) in Trieste, Italy.

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

References

1. LoRa Alliance^{TM}. What Is the LoRaWAN^{TM} Specification? Available online: https://lora-alliance.org/about-lorawan (accessed on 3 April 2019).
2. Semtech. Why LoRa? Available online: https://www.semtech.com/lora/why-lora (accessed on 3 April 2019).
3. LoRaWAN. Available online: https://www.thethingsnetwork.org/docs/lorawan/ (accessed on 3 April 2019).

4. TTN. THE THINGS NETWORK. Available online: https://www.thethingsnetwork.org/ (accessed on 3 April 2019).
5. ITU. ICT Facts and Figures. 2017. Available online: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf (accessed on 3 April 2019).
6. Kiruthika, R.; Umamakeswari, A. Low cost pollution control and air quality monitoring system using Raspberry Pi for Internet-of-Things. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 2319–2326. [CrossRef]
7. Ahmed, M.M.; Banu, S.; Paul, B. Real-time air quality monitoring system for Bangladesh's perspective based on Internet-of-Things. In Proceedings of the 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 7–9 December 2017; pp. 1–5. [CrossRef]
8. Chen, L.J.; Ho, Y.H.; Lee, H.C.; Wu, H.C.; Liu, H.M.; Hsieh, H.H.; Huang, Y.T.; Lung, S.C.C. An Open Framework for Participatory PM2.5 Monitoring in Smart Cities. *IEEE J. Mag.* **2017**, *5*, 14441–14454. [CrossRef]
9. Desai, N.S.; Alex, J.S.R. IoT based air pollution monitoring and predictor system on Beagle bone black. In Proceedings of the 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), Chennai, India, 23–25 March 2017; pp. 367–370. [CrossRef]
10. Kodali, R.K.; Mandal, S.; Haider, S.S. Flow based environmental monitoring for smart cities. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 455–460. [CrossRef]
11. Kumar, S.; Jasuja, A. Air quality monitoring system based on IoT using Raspberry Pi. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 1341–1346. [CrossRef]
12. Parmar, G.; Lakhani, S.; Chattopadhyay, M.K. An IoT based low cost air pollution monitoring system. In Proceedings of the 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, India, 27–29 October 2017; pp. 524–528. [CrossRef]
13. Barro, P.A.; Degila, J.; Zennaro, M.; Wamba, S.F. Towards Smart and Sustainable Future Cities Based on Internet-of-Things for Developing Countries: What Approach for Africa? *EAI Endorsed Trans. Internet Things* **2018**, *4*. [CrossRef]
14. THE WORLD BANK. Decline of Global Extreme Poverty Continues but Has Slowed: World Bank. 2018. Available online: https://www.worldbank.org/en/news/press-release/2018/09/19/decline-of-global-extreme-poverty-continues-but-has-slowed-world-bank (accessed on 3 April 2019).
15. Oniga, B.; Munteanu, A.; Dadarlat, V. Open-source multi-protocol gateway for Internet-of-Things. In Proceedings of the 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet), Cluj-Napoca, Romania, 6–8 September 2018; pp. 1–6. [CrossRef]
16. Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog Computing for Sustainable Smart Cities: A Survey. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 32. [CrossRef]
17. Pape A. BARRO. A Multi-Services and Multi-Optional Gateway for Communities. 2018. Available online: https://github.com/pape-barro/edge-gateway (accessed on 3 April 2019).