IADIS International Journal on Computer Science and Information Systems
Vol. 19, No. 1, pp. 48-64
ISSN: 1646-3692

PRACTICAL PROBLEMS AND SOLUTIONS TO HORIZONTAL INTEGRATION

Richard Dabels, Marvin Davieds, Frank Russow and Thomas Mundt
Department of Computer Science, University of Rostock, Rostock, Germany

ABSTRACT

The Internet of Things (IoT) is fragmented into many different smart environments, each requiring specific technical solutions for their use cases. This is a problem that has worsened over the years as new standards and technologies are constantly being developed. The term "horizontal integration" is used in literature to describe the attempt to reunite this fragmented IoT landscape. Abstract (in the sense of the OSI model) solutions are often proposed for this. However, the limitations of the underlying technologies are rarely sufficiently pointed out. This paper examines the integration of two smart environments as examples – the Smart Home and the Smart City. More specifically, it takes a theoretical look at how technologies can be connected between two Smart Home systems with the help of a Smart City technology and the problems that arise with it. ZigBee and LoRa are used for this purpose as exemplary technologies. Various possible application scenarios are shown to carry out a qualitative assessment of the feasibility of such a solution. The resulting problems of such an integration are shown and possible solutions for these are discussed.

KEYWORDS

Horizontal Integration, LoRa, ZigBee, Smart Environments

1. INTRODUCTION

The horizontal integration of smart environments is a major problem of the Internet of Things (IoT) (Al-Fuqaha et al., 2015; Filipponi et al., 2010; Prazeres and Serrano, 2016; Noura et al., 2019; Dabels, 2023). IoT as a term itself is also hard to define. Miorandi et al. refer to the IoT as a dynamic and distributed network of smart objects, also called things, that provide or consume data related to the physical world. Things are entities that have (1) certain physical characteristics (like shape and size), (2) communication functionalities, (3) a unique identifier, (4) a human-readable name that describes the object as well as a machine-readable address used to communicate with the object, (5) the ability to perform basic or even more advanced computational task, and finally are (6) able to sense or influence the physical reality around them (Miorandi et al., 2012). Here IoT is, among other things, characterized by its heterogeneity

in terms of the devices taking part in the systems as well as their computational and communication capabilities that necessitate bespoke architectures and protocols. Depending on the intended use, for example as a smart home, smart hospital, smart city, or other smart environment, a wide variety of technologies is used.

The fact that the term IoT is as complex as its users means that its landscape is extremely fragmented. This can be seen from the fact that each technology has its protocols, APIs, devices, and platforms, which are often incompatible with one another (Noura et al., 2019). By the horizontal integration of these smart environments, we mean overcoming this fragmentation and making the IoT systems more compatible with each other.

Related to IoT and horizontal integration is the topic of Society 5.0, which as a concept has been gaining traction over the last couple of years (Fukuyama, 2018; Deguchi et al., 2020; Mishra et al., 2022). It describes an iteration of concepts already known, like data collected from sensors – the "real world" as described by Deguchi et al. – being used and processed by computers. The difference with Society 5.0 is the impact this automated data processing has on society, actively guiding the decisions of its participants by "merging the physical space and cyberspace" (Deguchi et al., 2020).

But still, there is the problem of interoperability and fragmentation of IoT systems. Mishra et al. describe a smart city architecture consisting of four layers: The Sensing Layer, Data Management Layer, Transmission Layer, and Application Layer (Mishra et al., 2022). Of particular interest for this work are the Sensing and the Transmission Layer, since here the conflict of horizontal integration plays out. Both are responsible for accessing diverse technologies which creates a myriad of problems, some of which are addressed within this paper.

There are many approaches to solving this interoperability problem in the literature that implement middlewares with publish/subscribe mechanisms (Prazeres and Serrano, 2016; Filipponi et al., 2010; Collina et al., 2012; Dave et al., 2020). The Message Queuing Telemetry Transport protocol (MQTT) is often used and the projects are usually oriented towards the upper layer of the OSI model. There are also proposals on the opposite side, such as the use of software-defined radios, which can be used to implement a wide variety of technologies on the same hardware (Gavrila et al., 2018; Lin et al., 2013).

Another problem in the literature is that the definition of smart environments is very vague. In (Dabels, 2023), the term *Smart X* is introduced, which defines smart environments based on their technological characteristics. A smart home and a smart city thus differ in terms of the technologies and requirements placed on them and can therefore be considered distinct Smart X.

This paper examines the extent to which LoRa, a technology for Long Range communication within smart cities, can be suitable as a tunnel technology for ZigBee networks, which are commonly used in smart home systems. Since the smart city and smart home are understood as Smart X, the considerations should in principle also apply to other technologies in the respective areas.

In doing so, problems of the respective technology are addressed which, to the authors' knowledge, have received little attention in the literature to date. As soon as integrations are developed for the smart city, problems arise about the data throughput, latency, or the respective duty cycle. The latter in particular is a concept that is not generally known in conventional computer networks but represents a significant limitation in technologies such as LoRa (Adelantado et al., 2017).

Using network traffic from an existing ZigBee network, calculations are made to determine the feasibility of a ZigBee network that internally uses a LoRa bridge to overcome longer distances. Solutions are postulated for the associated problems as to how this bridging can be improved in the future. Limitations of the technologies that will inevitably arise when implementing a concept such as the horizontal integration of such fundamentally different technologies are also highlighted.

2. RELATED WORK

The problem of interoperability in IoT networks is by no means new. There are several approaches in the literature that deal with the problems of connecting distinct smart environments. This is often referred to as the horizontal integration of *vertices* or *horizontal markets* (Al-Fuqaha et al., 2015; Filipponi et al., 2010; Prazeres and Serrano, 2016; Noura et al., 2019; Dabels, 2023). An attempt to categorize smart environments is made in (Dabels, 2023), where the term *Smart X* is used to point out the technological differences between the various smart environments. The integration of such Smart X is intended to simplify and more precisely define the problem of horizontal integration.

To enable horizontal integration, middleware architectures are often used to implement a compatibility layer between the various networks and technologies. In (Prazeres and Serrano, 2016), for example, the concept of *fog computing* is applied to the Internet of Things. Here, the computing power of a network is shifted to its edge and closer to the information sources. This results in a platform called SOFT-IoT with a paradigm that Prazeres and Serrano call *Fog of Things*. The devices are connected via an MQTT middleware, while interfaces to other protocols and services are established via so-called *FoT gateways*.

Another project is *SOFIA*, which was developed as an *Event Driven Architecture* for the monitoring and management of smart cities (Filipponi et al., 2010). Here, networks consist of so-called *Semantic Information Brokers* (SIB) and *Knowledge Processors* (KP), which each implement publish/subscribe mechanisms and are responsible for the generation and utilization of information. Together they form a core network of SIBs, with the KPs representing the interface to the network. Among the smart environments mentioned are Smart Personal Spaces, Smart Indoor Spaces, and Smart City Spaces.

Another system for improving interoperability is the QUEST broker (Collina et al., 2012). This broker supports multiple protocols and stores data in its own format. Publish/subscribe mechanisms are supported by an MQTT server on each instance. The QUEST broker from Collina et al. also extends the MQTT server with a Representational State Transfer (REST) interface on which the most recently published values are provided.

Dave et al. have developed another broker called PONTE to improve interoperability. The focus of this project is on interoperability at the application layer, especially between MQTT, Constrained Application Protocol (CoAP), and HTTP (Dave et al., 2020). As PONTE is a further development of the aforementioned QEST broker (Collina et al., 2012), data persistence and an HTTP interface are supported in addition to the MQTT publish/subscribe capabilities. The enhancements compared to QEST essentially lie in the implementation of a CoAP server and other data storage engines such as MongoDB, LevelDB, and Redis.

The architectures just mentioned are just a small selection of the middleware solutions developed in the literature. According to (Derhamy et al., 2017), having a large selection of

solutions creates a new problem: middlewares also need translation mechanisms to be compatible with each other. The compatibility problem is therefore not solved but merely shifted to a more abstract level. Instead, a service-oriented translator is proposed by (Derhamy et al., 2017), with the help of which the integration competence is shifted to a participant of the in-house *Arrowhead* framework[1]. Among other things, it is worth mentioning that an intermediate data format is used for the translation process, and protocols are not translated directly into each other.

Interoperability is also established when specialized solutions are developed for the interconnection of two specific protocols. For example, Shi et al. shows how certain ZigBee control commands can be transmitted via LoRa (Shi et al., 2019). This works by selecting the LoRa user data in such a way that it is interpreted by ZigBee devices as valid signals (payload encoding). In this way, a direct connection between devices can be established without routing via the mesh network.

However, solutions on lower layers of the network stack are also possible. Software-defined radios are theoretically capable of being configured for a wide range of IoT protocols. Universal network devices that can be dynamically expanded to include new radio technologies are therefore conceivable. Projects on this topic have already been developed, for example, (Gavrila et al., 2018) and (Lin et al., 2013).

Finally, protocol convergence is also a possibility for the future of the IoT. For this consideration, Derhamy et al. compares the development of the IoT with that of the Internet, where convergence towards the Internet Protocol has taken place (Derhamy et al., 2017). However, it is also noted that beyond TCP and UDP, there are still many protocols that are not compatible with each other.

3. SMART HOME TO SMART CITY

The test scenario described in this paper represents the integration between Smart Home and Smart City. Smart Home and Smart City are Smart X as described by (Dabels, 2023). This means that the research presented here can, in principle, also be translated to other Smart Home and Smart City technologies. The test setup consists of an existing Smart Home ZigBee network and the study is strictly theoretical. The data recorded will be used to determine the extent to which LoRa is suitable for tunneling ZigBee traffic between two networks.

A realistic scenario for such a test setup would be, for example, a flood or irrigation sensor in a secondary ZigBee setup that notifies the user of the system in their home of an incident in a poorly connected location via the primary ZigBee network. This can be seen in Figure 1. Please also note that in this scenario, we do not have two independent networks, but one network which is extended by a LoRa tunnel.

However, this general test setup itself leaves many implementation details open, which will affect the network parameters discussed below. For example, LoRa is usually only used in so-called LoRaWANs. This means that sensor data is not transmitted directly between two end devices but is first sent via a LoRaWAN network with gateways, network, and application servers. Although point-to-point connections are possible with LoRa, they are not representative of a realistic application scenario.

[1] https://arrowhead.eu

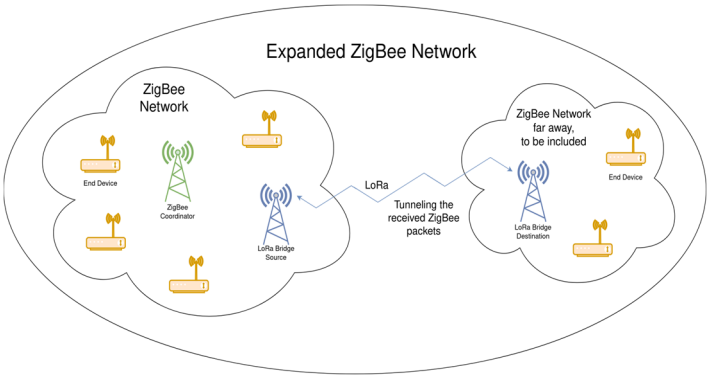

Figure 1. Extended ZigBee network

There is also the question of how the ZigBee sensor is connected to an actuator. Most of the literature describes middleware architectures that can be used very flexibly (Prazeres and Serrano, 2016; Filipponi et al., 2010; Collina et al., 2012; Dave et al., 2020). However, it is also theoretically possible to tunnel ZigBee devices via a technology such as LoRa and thus establish a direct connection between sensors and actuators. In the application scenario of Figure 1, this is even possible without using sequence numbers such as in TCP to reestablish order, as the maximum size of a LoRa telegram is greater than that of ZigBee (IEEE Computer Society, 2009; Semtech Corporation, 2019).

Where possible, such special features are addressed and presented within this paper. Nevertheless, it cannot be guaranteed that the scenario described here is equivalent to a real application of the technologies. Many possibilities for optimizing such a mechanism for data compression are also deliberately omitted.

For the experiments presented here, no ZigBee tunnel was implemented via LoRa. Instead, representative ZigBee data streams were recorded, and calculations were performed with the help of the respective documentation (IEEE Computer Society, 2009; Semtech Corporation, 2020; Semtech Corporation, 2019) as well as other sources. With the help of these calculations, we can make qualitative statements about the methods of the LoRa tunnel.

4. EXPERIMENTAL SETUP

The test setup consists of an existing ZigBee network. A Conbee 2 with the ZSHARK sniffing firmware from dresden elektronik[2] was used to record the data. With this firmware, it is possible to read and save the ZigBee network traffic in Wireshark, a tool commonly used in network analysis. The ZigBee network key known in advance was stored in Wireshark to be able to analyze the data traffic in plain text. Additionally, a SONOFF ZigBee 3.0 USB Dongle Plus is used as a coordinator for the existing ZigBee network. For the Smart Home system, an instance of Home Assistant was used, whereby ZigBee2MQTT is used to manage the network.

[2] https://www.dresden-elektronik.com/wireless/software/zshark.html

From the recorded data traffic, only the sections that are representative of one of the following three scenarios are generated with the help of filters. These scenarios are intended to show the applications for which a ZigBee tunnel via LoRa would be suitable in a real scenario and are as follows:

1. *RFD Join:* The join process is essential for a functional ZigBee tunnel – if no join can be performed via it, devices can only be integrated into the network by physically transporting them to the coordinator's network. The abbreviation RFD stands for *Reduced Function Device*, which is a device class in ZigBee alongside the *Full Function Device*. The main difference between these classes is whether the devices are battery-operated and therefore not routing-capable. The packets that are required for a successful join process are selected by a suitable filter within Wireshark. This includes the *Beacon Request* and *Association Request* of the RFD, the *Beacon* of the router to which the RFD has connected, the *Association Response* with which the RFD is assigned a network address and all subsequent packets in which the RFD appears as the source or destination address. The latter are sent out for further configuration and to read out the capabilities of the new device. As no assumptions should be made in this work about the relevance of individual broadcasts for the RFD, all broadcasts in the ZigBee network are also transmitted via the tunnel as soon as a short address has been assigned.
2. *Single Device:* Only the data traffic of a single RFD device is considered in this scenario. It is representative of the setup shown in Figure 1 – for example, a flooding or temperature sensor that is to be connected to a ZigBee home network over a greater distance using LoRa. Of all three scenarios described, the lowest traffic volume is to be expected in this scenario, as only one device and the coordinator are involved in the communication. All broadcast messages are also included here for a worst-case estimate.
3. *No Filters*: This scenario shows how the LoRa tunnel would react under the total traffic volume of a ZigBee network in a small apartment with 13 FFDs and 11 RFDs. This is a worst-case scenario insofar as the entire traffic of an apartment’s network is transmitted via the tunnel.

5. PERCEIVED PROBLEMS

In this section, various problems are determined that can occur under the test setup proposed in Figure 1. Values from the official standards for maximum packet sizes and from the literature are used for the calculations. Examples of this can be maximum packet sizes or an expected runtime for protocol translation.

5.1 Latency

Latency is probably one of the most noticeable problems for the end user. It is not only the line-up latency in complex integrated networks that is a problem, but also the translation of protocols into each other. How long the input delay can be, depends largely on the user and the usage environment. Residents of a Smart Home will generally have certain expectations on the speed at which their input is executed. Mozilla states that input latency should be less than 100

ms for websites[3]. As smart homes are a completely different concept, this is only a guideline. Realistically, input delays of up to 200ms should also be acceptable, although this also depends on the application. This value is also recommended by Google for the smart home sector[4].

However, the requirements for other smart environments, such as Smart Grids, are much more significant. Changes in the power grid must be responded to within a very short time, which is a major challenge, especially in times of the energy transition. In comparison with the Smart Home, however, the line-up latency of the individual nodes of the Smart Grid must be taken into account, as more communication participants are to be expected between the transmitter and receiver than in Smart Homes. A Smart Grid with a network structure comparable to a smart home should generate significantly lower latencies.

Admittedly, the latency for a use case such as in Figure 1 is not particularly important: For a flood warning, it is not relevant to a resident whether they are notified within one or ten seconds. For the general use case, however, a relatively low latency is desirable. The reason for this is that the horizontal integration of smart environments should not be considered based on individual cases, but universally applicable concepts should be developed. A solution that promises latencies of less than 10 seconds will not be sufficient for many smart environments. An extreme case occurs when the duty cycle of such a system is exceeded. This is discussed in more detail in Section 5.4.

In Section 4 an experimental setup is described in which two LoRa nodes communicate directly with each other. It was also mentioned that this is not a common application for LoRa, as this is normally used within a WAN in LoRaWAN. Potsch and Hammer also deal in detail with the problem of latency in LoRaWAN networks (Potsch and Hammer, 2019). It turns out that the latency in LoRaWAN networks is mainly dependent on the spreading factor (SF) and the frequency used. Even the transmission of just a few bytes (8) with a high SF (12) at 125 kHz generates latencies of more than one second for the airtime of the telegram alone. With 50 bytes and SF 12, this is already around 2.5 seconds. In a LoRaWAN network, on the other hand, latencies of less than one second from one node directly to another cannot be achieved even under optimum conditions.

Potsch and Hammer conclude that LoRaWAN is not suitable for Industrial IoT (IIoT) scenarios if processes must be carried out in real-time. Instead, it is explicitly pointed out that less time-critical processes such as smart metering and building automation are feasible. For the test setup shown in Section 4, the delays mentioned are acceptable.

5.2 Protocol Interoperability

For different protocols to be translated to each other, there must be a cross-system semantic understanding of the communicated data. In (Rahman and Hussain, 2020), various problems associated with semantic interoperability in the IoT are highlighted. However, it is proving very difficult to find a universally valid standard that can be applied across the board. Works such as those by (Dave et al., 2020) and (Collina et al., 2012) even define middleware architectures that deal with the problem of interoperability. However, the numerous middleware solutions that already exist in the literature do not so much solve the actual problem of interoperability as shift it to interoperability between the individual middlewares (Derhamy et al., 2017).

[3] https://developer.mozilla.org/en-US/docs/Web/Performance/How_long_is_too_long
[4] https://developers.home.google.com/cloud-to-cloud/support/faq#response-latency

This paper does not propose a universal solution. The use of ZigBee and LoRa results in some special features that cannot be transferred to other interoperability problems. For example, if two ZigBee networks are to be connected, the protocols do not necessarily have to be translated. A ZigBee packet is up to 127 bytes in size (IEEE Computer Society, 2009), while the maximum size of a LoRa packet is 255 bytes (Semtech Corporation, 2019). This makes it possible to tunnel ZigBee traffic through LoRa traffic, even if other approaches could be more efficient.

In reality, however, it cannot be assumed that tunneling between different protocols is possible. It is precisely for such cases that concepts must be developed as to how protocols can be translated into each other. In (Noura et al., 2019) a taxonomy is presented that better breaks down the problem of interoperability. It mentions 5 different classes of interoperability:

The so-called **device interoperability** deals with the interplay of different end devices, which also differ in terms of their communication protocols. Gateways are often used between the respective devices to establish this interoperability. **Network interoperability** describes how communication between different networks can be achieved. This is very complex in the IoT sector, as it is made up of a wide variety of heterogeneous networks. In this "network of networks", problems such as addressing, routing and quality of service must also be taken into account. The **syntactical interoperability** describes that the messages sent to each other follow a standardized format (i.e. XML or JSON) and that interfaces (i.e. REST) are defined for all resources. Another important aspect is **semantic interoperability**, which focuses on the interpretation of the data sent. Common examples of this are the different representations of measurement units, such as Celsius and Fahrenheit, or meters and miles, and the problems that an incorrect interpretation of these values can cause. The last interoperability is **platform interoperability** which is caused by the numerous environments and operating systems of the individual manufacturers, which prevent further integration.

The integration of ZigBee and LoRa presented here can also be classified in this taxonomy. Of particular relevance for this paper is the **network interoperability** and the **device interoperability** since we aim to "enable seamless message exchange between systems through different networks for end-to-end communication" (Noura et al., 2019); the different networks being ZigBee and LoRa networks and the end-to-end communication between two ZigBee devices.

Noura et al. explicitly proposes solutions for **network interoperability** in (Noura et al., 2019). The solution used in this paper is essentially that of an adapter, even if, thanks to tunneling, neither syntactic nor semantic information needs to be translated into another format.

There are some parallels with the work of (Arzo et al., 2021), which also deals with *network interoperability*, but in this case between LoRaWAN, Wi-Fi and 6LoWPAN. The architecture proposed there uses SDRs to ensure basic compatibility with various radio technologies at the device level. At the network level, software-defined networking is used and exposed via the proposed architecture. With regard to the use of an IP backbone, this approach is similar to most of the projects mentioned in Section 2. This is where the biggest differences between the concept presented in (Arzo et al., 2021) and most existing solutions lie, as tunneling also enables integration without using the Internet.

In the long term, however, problems must be solved at all of the aforementioned levels. The advantage of our approach is that the integration of smart environments is no longer dependent on an IP-based backbone. A disadvantage, on the other hand, is that an approach via adapters entails considerable problems with regard to scalability (Noura et al., 2019). These problems can be largely circumvented in this work by tunneling.

5.3 Data Throughput

Establishing an end-to-end connection via a tunnel or similar method is only one prerequisite for successful communication. If an overload occurs in conventional IP networks, there are various ways of throttling the data rate. These include, for example, dropping packets, which reduces the window size in TCP and thus the data rate. Overloads occur when a participant in the network traffic can no longer process all packets in real-time.

Although the tunneling of ZigBee packets via LoRa can work, data throughput is a major problem. The bit transmission rate depends on the frequency used and the Spreading Factor (SF), of which there are six (SF7-SF12) to choose from. One disadvantage of using a low SF is that the distance over which the technology can be used to communicate is reduced. The upside however is that the data rate is much greater using a low SF.

Depending on these parameters, data rates of 0.3 kbit/s to 27 kbit/s are possible (Adelantado et al., 2017). However, this does not refer to the usable data rate, but to the gross or raw data rate. Only 0.3 kbit/s can be achieved if SF12 and a frequency of 125 kHz are set. The highest possible data rate of 27 kbit/s can be achieved when transmitting under SF7 at 500 kHz. According to (Goursaud and Gorce, 2015), the following formula can be used to calculate the data rates:

$$R = SF \cdot \frac{BW}{2^{SF}}$$

As a frequency of 500 kHz is not permitted in Europe according to (ETSI, 2018), 250 kHz is used for the following calculations. SF7 is also assumed to represent the best-case scenario for LoRa. Even under these assumptions, data transmission with LoRa is many times lower than in ZigBee. At full capacity, a LoRa network will therefore never be able to tunnel the entire network traffic of ZigBee.

RF Join: The entire pairing process took 19.6 seconds in the test setup from Section 4 and according to Wireshark, 10,297 bytes (82,376 bits) were transmitted. The effective transmission rate of 4.2 kbit/s is far below the maximum and far above the minimum data rate of LoRa. Under the right conditions, the data rate for pairing should therefore not be problematic. The cutoff is, for example, at SF9 and 250 kHz (4.4 kbit/s) or SF7 and 125 kHz (6.8 kbit/s). However, as the transmission time of LoRa and the retransmission of ZigBee traffic on the other side of the tunnel must be added together, timing problems may occur during pairing. In a realistic deployment scenario with LoRaWAN, this line-up latency accounts for a large part of the overall latency (Potsch and Hammer, 2019). The concerns regarding successful pairing processes can neither be dispelled nor confirmed by the purely theoretical consideration of this process.

Single Device: The device selected in this experiment can measure three data points. Depending on the time or sufficiently large value change, these are sent individually or summarized in *Report Attributes* commands from the sensor. Seven such packets with an average size of 65 bytes were captured during a recording which took place in the morning between 6:00 and 7:30 and lasted 5739 seconds to represent a typical morning routine. This corresponds to a data transmission rate of 0.6 bit/s, which can even be achieved under the worst-case conditions for LoRa (SF12 with 125 kHz, 0.37 kbit/s). It should therefore be possible to tunnel individual devices.

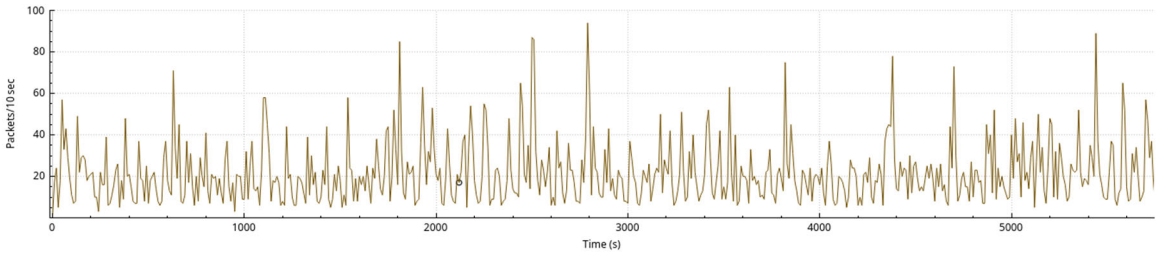

Figure 2. Wireshark I/O Graph for non-filtered data

No Filters: A total of 737,008 bytes (5,896,064 bits) were transmitted in the same duration, which corresponds to a data transmission rate of 1.027 kbit/s. This value is also far below the maximum, but still above the minimum data transmission rate of LoRa and could even be achieved with SF10 at 125 kHz (1.2 kbit/s) or SF11 at 250 kHz (1.3 kbit/s). However, problems could occur if the data rates in a smaller time window are higher than the average data rate over the entire period under consideration. In Figure 2 it can be seen that the use of the ZigBee network was relatively even over the recording window and no particular load peaks occurred. In principle, however, the combined use of smart home functions may generate data volumes that temporarily overload the data rates enabled by LoRa. This was already the case in RF Join, for example. Theoretically, however, the data transmission rate should only pose a problem under the most unfavorable conditions possible.

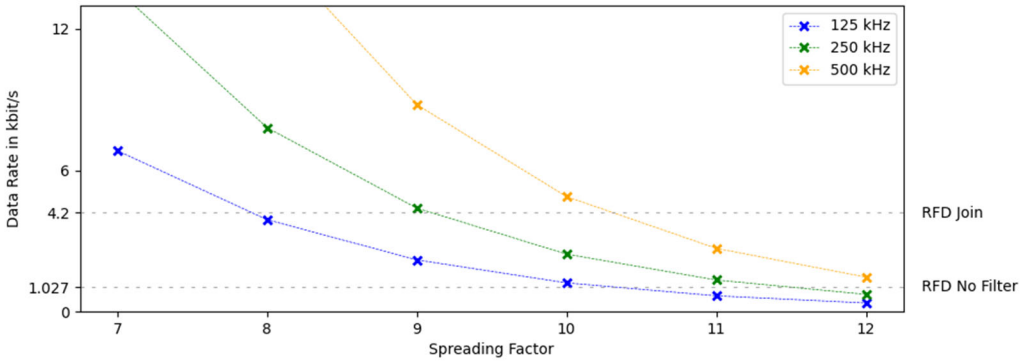

Figure 3. LoRa data rate depending on the spreading factor and bandwidth

We can conclude that the sufficiency of the data rate is highly dependent on the parameters of LoRa as well as the workload of the bridged ZigBee network. Figure 3 shows those parameters for each of the scenarios mentioned. If peaks like in the RFD Join scenario appear, most setups fail to deliver a sufficient data rate. In case no such peaks appear during regular traffic, the data rate is not enough under very high SF and 125 kHz. A tunnel for only a single device although has such low requirements (0.6 bit/s), that we chose not to render it within Figure 3.

5.4 Duty Cycle

Duty cycles are a mechanism that primarily occurs in IoT networks and is used to regulate data traffic in media that are used by many participants. They define the so-called *relative frequency occupancy time* and thus determine how long the devices in a network are allowed to send packets within a certain time window. An alternative to duty cycles are the so-called *listen before talk* mechanisms, which are often not implemented in IoT networks due to the higher energy restrictions. Conventional networks based on IEEE 802.3 (Ethernet) or IEEE 802.11 (WiFi) implement different mechanisms for collision avoidance.

The duty cycle for LoRa is defined in the ETSI standard EN 300 220-2 and is set at < 1 % per hour for most channels (ETSI, 2018). If this value is reached by transmissions within the network, the corresponding device must not make any further transmissions. In normal use, it is rare for the duty cycle to be reached in LoRaWAN networks. However, the tunneling of packets through LoRa is also not an intended use of the technology and may very well exceed this value.

The Things Network provides an online tool for calculating airtime[5]. For the best-case scenario, this means that 222 bytes of user data can be transferred in 184 ms. This average value is then used to calculate the airtime and duty cycle.

RF Join: During pairing, 10,297 bytes were transmitted in 247 packets under the conditions specified in Section 4. The airtime of the transmission of the entire process at SF 7 and 250 kHz amounts to 6.02 seconds. Of the maximum 36 seconds of the duty cycle, 16.73 % is therefore used. This means that five pairing processes can take place per hour under optimum conditions. Figure 4 shows the amount of time a pairing process would take and thus how much of the duty cycle will be used depending on the data rate used by the LoRa tunnel.

Single Device: As described above, we recorded seven data packets with an average payload size of 65 bytes for the selected RFD. This results in a total average airtime of $61.57\ ms \cdot 7 = 430.99\ ms = 0.43\ s$, which is well below the maximum airtime of 36 s at < 1 % channel utilization. Accordingly, individual RFDs with a low transmission frequency are suitable for tunneling via LoRa, even when using only one channel.

No Filters: Over the entire recording interval of 5739 seconds, 737,008 bytes of data were transmitted in 12471 messages during normal operation. This means that the packets have an average payload size of 59 bytes. If this network traffic were transmitted via LoRa, the average airtime per data packet would be 56.4 ms when using SF7 and 250 kHz and the total airtime for all data packets recorded and tunneled via LoRa would be 703,3644 s. Since only 36 s of airtime can be allocated for a duty cycle of < 1 %, this amount of data cannot be transmitted in 1.5 h under any circumstances. It is therefore clear that an average ZigBee network cannot be completely tunneled in practice.

[5] https://www.thethingsnetwork.org/airtime-calculator/

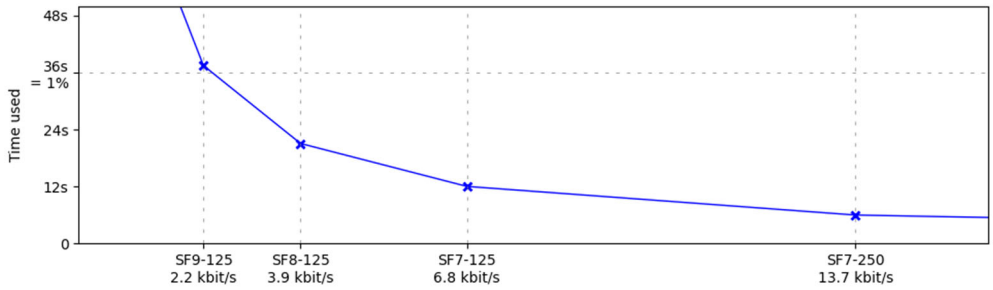

Figure 4. LoRa data rate depending on the spreading factor and bandwidth

6. FURTHER POSSIBLE OPTIMIZATIONS

Various problems that occur during horizontal integration have been discussed in Section 5. This section focuses on the potential for the optimization of the approach developed within this paper. In particular, we take a look at the effect of filtering data before transmission on the LoRa tunnel, as well as data selection. Both aim to reduce the amount of traffic to be transmitted. Lastly, we go into detail on the possibility of using frequency hopping to further extend the available airtime of the LoRa tunnel.

6.1 Filtering

The problems discussed so far can be mitigated by reducing the actual data traffic via the tunnel. This can be achieved by filtering out messages that are not required on the side of the *LoRa Bridge Destination* (see Figure 1). For this purpose, it is conceivable to keep a table on the *LoRa Bridge Source* that contains all relevant information about the devices on the other side. This includes the MAC address, network address and the type of device. The following rules are now defined, according to which the *LoRa Bridge Source* selects whether a packet is routed through the tunnel:

1. If the packet is an *IEEE 802.15.4 Beacon Request*, it is not forwarded.
2. If the *ZigBee Network Layer Destination Address* is **0xfffd** or **0xfffc** and there is at least one router on the destination side, it will be forwarded.
3. If the *ZigBee Network Layer Destination Address* is equal to **0xfffd** or **0xffff**, it is forwarded.
4. The IEEE 802.15.4 MAC Destination Address belongs to one of the devices connected via the tunnel.

The first rule prevents devices on the source side from connecting to routers on the destination side and thus generating unnecessary data traffic and latencies via the tunnel. Rule 2) reduces data traffic through broadcasts (zigbee Alliance, 2017) by only transmitting broadcasts that are only directed to routers (with the destination addresses **0xfffb** and **0xfffc**) via the tunnel if there is actually at least one router on the destination side. All other broadcasts for all devices (**0xffff**) or all non-sleeping devices (**0xfffd**) are always forwarded using rule 3). All other packets are forwarded using rule 4) if the destination address of the packet belongs to

a device on the destination side of the tunnel. Packets that arrive on the destination side are first transmitted through the tunnel to the source side without any restrictions.

This principle could even be transferred to group messages, which are sent out as broadcasts and addressed to the respective group or its members via the group ID in the *Application Support Sublayer*. If the *LoRa Bridge Source* is aware of the group membership of the devices that are connected to the ZigBee network via the tunnel, group messages for which there is no suitable recipient behind the tunnel can be filtered out.

6.2 Packet Selection

In addition to the stateless filtering of packets, we see further data saving potential by summarizing and preselecting certain messages. However, this requires a closer look at individual processes in the network or commands of specific device classes or individual devices and cannot necessarily be generalized for other device classes and processes.

An example of the selection from several messages can be seen in the pairing process. A device initiates the pairing process by sending a *Beacon Request*. Routers then send out a *Beacon* with the *Extended PAN ID* assigned to their Personal Area Network (PAN) and information on whether a join is permitted. After sending a *Beacon Request* from the other side of the tunnel, the *LoRa Bridge Source* can now collect the *Beacon* packets and, after a certain waiting time (in our recordings, the beacons of all routers were received after 100 ms at the latest), select a beacon – for example, according to the best perceived transmission power – and transmit only this one via the tunnel. As a result, the device at the other end of the tunnel will only receive the selected beacon and want to associate with the corresponding router with an *Association Request*. This prevents *Beacon* packets have to be transmitted from each router via the tunnel.

6.3 Frequency Hopping

In the ETSI standard EN 300 220-2 (ETSI, 2018), the LoRa frequency range is divided into six sub-ranges in which a transmission power of 25 mW is permitted. A duty cycle of 1 % applies in the three largest frequency ranges, 0.1 % in two others and even 10 % in a range of just 250 kHz. The duty cycle only applies to a sub-range and not to the entire LoRa band. This means that as soon as the duty cycle of 1 % in the 865 MHz to 868 MHz range has been reached, the system can switch to another frequency range, such as 868.0 MHz to 868.6 MHz, to continue transmitting.

To ensure that both end points of the LoRa tunnel transmit and listen on the same frequency, a synchronization procedure must be developed for frequency hopping in order to enable permanent data exchange. A simple solution would be for both devices to start communicating on a fixed frequency and to signal a change in frequency with a specific data packet using the master-slave principle.

If all partial frequency ranges are fully utilized, a duty cycle of 13.2 % can be achieved. This means that an airtime of up to 475.2 seconds can be used with one device. With a spreading factor of 7 and a bandwidth of 250 kHz, which is also permitted in the partial frequency range with a duty cycle of 10 %, this corresponds to a data volume of 653,400 bytes.

In the following, the results from the experiments in Section 5.3 are taken up again and reconsidered with a duty cycle of 13.2 % using SF 7 at 250 kHz for transmission:

RFD Join: 10,297 bytes were transmitted for one pairing process. While five pairing processes were possible within one hour with a duty cycle of 1 %, this number rises to 63 when using all partial frequency ranges.

Single Device: With the selected sensor, seven data packets with a total size of 455 bytes were measured over a period of 5739 seconds. Calculated to a timespan of one hour, this corresponds to a data volume of 286 bytes. With a possible data volume of 653,400 bytes, the LoRa tunnel would be able to transmit the data volume of a total of 2883 such sensors without exceeding the duty cycle of 13.2 %. Figure 5 shows the maximum number of sensors that can be transmitted within the duty cycle, depending on some selected data rates.

No Filters: In this scenario, a total of 737,008 bytes were measured in the measurement period. Calculated to a timespan of an hour, the data volume is 462,316 bytes, which is still below the maximum possible data volume of 653,400 bytes and could therefore still be transmitted via the LoRa tunnel if all partial frequency ranges are used. However, if the network is expanded over time to include more sensors, it may no longer be possible to transmit this amount of data, so the aspects from Section 6.1 and Section 6.2 should still apply.

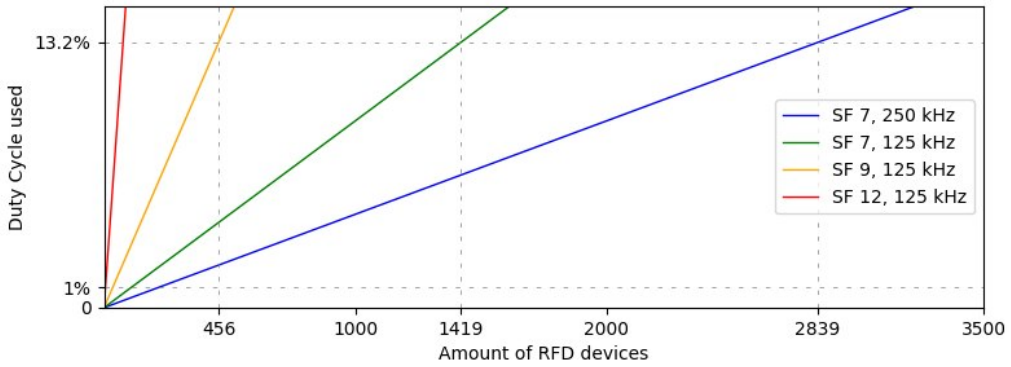

Figure 5. LoRa data rate depending on the spreading factor and bandwidth

7. CONCLUSION

We have shown several different problems that occur when integrating Smart Home and Smart City technologies. The protocols in question were ZigBee and LoRa – both of which are currently used within real application scenarios. Integrating these technologies may well be a relevant problem within the topic of horizontal integration (and in extension Society 5.0), in which we try to overcome the technological boundaries of different Smart X.

To this account, we have shown the limits of the technologies in question and examined four different concrete problems in view of an experimental setup which has been described in Section 4. The perceived problems mentioned in Section 5 are latency, protocol interoperability, data throughput, and duty cycle. We conclude that overcoming and addressing these problems is highly dependent on the technologies to be integrated and the parameters of their operation.

LoRa as an exemplary technology for Smart City alone shows how a system's suitability for integration purposes can vary based on its mode of operation. Figure 3 in particular shows that there is not one answer to whether these two technologies can be horizontally integrated, but

that it depends on multiple factors and the respective operational scenario. Moving forward, such parameters must be put into consideration when the question integrated smart environments arises.

This work shows only a selection of problems that can occur in horizontally integrated environments and approaches them theoretically. We fully expect more problems to arise when implementing concrete and low-level solutions to protocol interoperability and integration. Considering existing solutions mainly focus on a high-level approach to solving interoperability within smart environments, this can be considered an attempt to steer the discussion towards the inherent properties associated with technologies such as ZigBee, LoRa and many more.

8. FUTURE RESEARCH

The concept presented in this paper does not claim to integrate Smart X Smart Home and Smart City in a significantly efficient way. Rather, it aims to show how two disparate technologies can be transparently connected without the use of middleware. It has been shown that this is often not possible without compromise.

In Section 5.4, for example, no transmission of the entire network traffic is possible. Pairing is possible, but three times per hour at most. This may be sufficient for connecting a few sensors; however, failed attempts have an even greater negative effect. One reason for the rapid exhaustion of the duty cycle is that ZigBee data packets are transmitted in their entirety.

One solution is to aggregate the data before transmission. For example, a temperature sensor could only transmit an average value every 10 minutes rather than every minute. The data volume of individual connections could thus be optimized.

We see further potential for optimization in filtering the packets to be transmitted through the tunnel. For example, broadcasts such as *Permit Join Request* or *Device Announcement* could be aggregated again by routers and thus only routed through the tunnel once. If there are no routers on the side of the tunnel that is not home to the coordinator, packets relevant for routing could be filtered.

Another solution would be to encode and transmit the user data more efficiently. This would require an intermediate representation, which has already been developed in various projects. The disadvantage of such a solution would be that the complete transparency of the approach presented here would be lost.

The main problem in Section 5.3 is also the amount of data that must be transmitted via the slower LoRa tunnel. If the transmissions accumulate, this can lead to unexpected behavior in the communication. This may need to be investigated more closely in an implementation of the approach. In principle, both approaches mentioned above are also suitable for reducing the problem of data throughput.

To maintain the duty cycle, it would be possible to switch frequencies when a frequency is exhausted. A similar process is already used in Bluetooth under the name *frequency hopping*. Here, however, it is used more to avoid interference in the 2.4 GHz band than to use duty cycles multiple times. Such approaches are also conceivable in LoRa (Adelantado et al., 2017).

The combination of many heterogeneous technologies also raises questions about how such complex systems can be maintained. Auditing IoT networks is often a difficult process due to the focus on the energy efficiency of the sensors. In an integrated IoT landscape, this process becomes very difficult if it is made up of an extremely large number of subsystems.

Finally, solutions should also be developed for problems such as packet loss. LoRa already offers protection mechanisms if transmissions fail through the use of acknowledgments. However, the existence of such procedures cannot be assumed for every technology. For this reason, these and other problems addressed in this section should be considered in further work.

REFERENCES

Adelantado, F. et al., 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, Vol. 55, No. 9, pp. 34-40.

Al-Fuqaha, A. et al., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 4, pp. 2347-2376.

Arzo, S. et al., 2021. 'A Translator as Virtual Network Function for Network Level Interoperability of Different IoT Technologies', in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*. Tokyo, Japan, pp. 416-422

Collina, M. et al., 2012. 'Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST', in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, Sydney, Australia, pp. 36-41.

Dabels, R., 2023. 'Smart X: A Description for Smart Environments', in *e-Society 2023*. Lisbon, Portugal, pp. 1-8.

Dave, M. et al., 2020. 'Ponte Message Broker Bridge Configuration Using MQTT and CoAP Protocol for Interoperability of IoT', in *First International Conference, COMS2 2020*. Gujarat, India, pp. 184-195.

Deguchi et al., 2020. *Society 5.0: A people-centric super-smart society*. Springer Singapore, Singapore.

Derhamy, H. et al., 2017. IoT Interoperability – On-Demand and Low Latency Transparent Multiprotocol Translator. *IEEE Internet of Things Journal*, Vol. 5, No. 4, pp. 1754-1763

European Telecommunications Standards Institute, 2018. *EN 300 220-2, V3.2.1*

Filipponi, L. et al., 2010, 'Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors', in *2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*. Venice, Italy, pp. 281-286.

Fokuyama, M., 2018. Society 5.0: Aiming for a New Human-Centered Society. *Japan Spotlight*, Vol. 27, No. 5, pp. 47-50.

Gavrila, C. et al, 2018. 'Reconfigurable IoT Gateway Based on a SDR Platform', in *2018 International Conference on Communications (COMM)*. Bucharest, Romania, pp. 345-348.

Goursaud, C. and Gorce, J. M., 2015. Dedicated networks for IoT: PHY / MAC state of the art and challenges. *EAI Endorsed Transactions on Internet of Things*, Vol. 1, No. 1, pp. 1-11.

IEEE Computer Society, 2009. *IEEE Std 802.15.4™-2015*, IEEE Standard for Low-Rate Wireless Networks.

Lin, Y. et al., 2013. 'Wireless IoT Platform Based on SDR Technology', in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. Beijing, China, pp. 2245-2246.

Miorandi, D. et al., 2012. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, Vol. 10, No. 7, pp. 1497-1516.

Mishra, P., Thakur, P. and Singh, G., 2022. Sustainable Smart City to Society 5.0: State-of-the-Art and Research Challenges. *SAIEE Africa Research Journal*, Vol. 113, No. 4, pp. 152-164.

Noura, M. at al., 2019. Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mobile Networks and Applications*, Vol. 24, No. 3, pp. 796-809.

Potsch, A. and Hammer, F., 2019. 'Towards End-to-End Latency of LoRaWAN: Experimental Analysis and IIoT Applicability', in *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*. Sundsvall, Sweden, pp. 1-4.

Prazeres, C. and Serrano, M., 2016. 'SOFT-IoT: Self-Organizing FOG of Things', in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. Crans-Montana, Switzerland, pp. 803-808.

Rahman, H. and Hussain, Md. I., 2020. A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges. *Transactions on Emerging Telecommunications Technologies*, Vol. 31, No. 12, pp. 1-25.

Semtech Corporation, 2019. *LoRa and LoRaWAN: A Technical Overview.* https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf

Semtech Corporation, 2020. *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver.*

Shi, J., Mu, D. and Sha M., 2019. 'LoRaBee: Cross-Technology Communication from LoRa to ZigBee via Payload Encoding', in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. Chicago, IL, USA, pp. 1-11.

Zigbee Alliance, 2017. *Zigbee Specification, Revision 22.1.0*