A Methodology to Automate the Selection of LPWA Technologies in WSN Applications

Carlos Nascimento, António Casimiro, and Pedro Ferreira

LASIGE, Faculdade de Ciências da Universidade de Lisboa
cnascimento@lasige.di.fc.ul.pt, {casim,pmf}@ciencias.ulisboa.pt
http://www.lasige.di.fc.ul.pt

Abstract. The availability of various wireless communication technologies that may be used to implement a Wireless Sensor Network (WSN) allows system designers to select the one, or ones, that better satisfy application requirements. This selection process is usually done in a adhoc way, weighting the advantages and disadvantages of different possible solutions with respect to some requirements, often using empirical knowledge or simply dictated by the designer's preference for some particular technology. When several functional and non-functional requirements have to be addressed, finding an optimal or close to optimal solution may become a hard problem. In this paper we propose a methodology for addressing this optimization problem in an automated way. It considers various application requirements and the characteristics of the available technologies (including Sigfox, LoRa, NB-IoT, ZigBee and ZigBee Pro) and delivers the solution that better satisfies the requirements. We illustrate how the methodology is applied to a specific use case of WSN-based environmental monitoring in the Seixal Bay.

Keywords: LPWA · WSN · Optimization · Configuration · Methodology

1 Introduction

With the introduction of LPWA (Low Power Wide Area) technologies the available scope of wireless technologies for the deployment of WSN (Wireless Sensor Network) increased. There are now diverse technologies to choose from, which have different characteristics and differ at various technical levels. Where some technologies excel, others present limitations. This is because different technologies were created to be used in different situations and to address different application requirements.

Therefore, there is no single technology that is appropriate to address the outstanding requirements of every possible use case. Moreover, selecting one or a set of technologies that will be appropriate to address the application requirements is a hard problem. This is not only because the amount of possible options tends to be large, namely when considering possible combinations of technologies to serve in different parts of a WSN, but also because there are situations in which it is not obvious that one solution might be better than another, and, finally, because there are human aspects that often influence the selection process based only on empirical knowledge that is sometimes not substantiated. For instance, managerial decisions or personal bias can influence the selection process leading to either suboptimal or downright wrong selection choices. In this paper we propose a solution to address this problem. More precisely, we introduce a methodology to automate the process of selecting one or multiple LPWA technologies while satisfying a set of application requirements. By automating the process, we not only facilitate the selection process but also provide a way to reach optimal or close to optimal solutions, removing subjectivity, personal bias, and possible assessment mistakes from the technology selection process.

Our proposed methodology consists of four steps. They go from the gathering of information regarding the application requirements (e.g., concerning functional aspects like throughput or communication distance as well as non-functional aspects like cost or network homogeneity) to the provision of a set of possible solutions for the problem, including two steps whose purpose is to eliminate solutions that do not meet requirements. We argue that this solution hits the sweet spot in solving the problem of finding the right solutions for the network technologies to be used, which satisfy the application requirements and, whenever possible, still allow some degree of flexibility in the final choice.

The paper structure is as follows. Firstly, Section 3 provides a brief introduction to the LPWA technologies, to ZigBee and to ZigBee PRO, which are those considered in this work. A comparison of LPWA and ZigBee technologies is also provided. In Section 4, the application requirements used in the methodology are defined. The methodology itself is explained in Section 5, each sub-section describing one step in detail. Finally, in Section 6 we refer a real use case where we plan to apply the methodology, showing that it is indeed helpful and effective.

2 Related Work

Several works have been done regarding the development of methodologies applicable to WSNs.

In [6] and [8] different application use cases are identified and a taxonomy is established for the network requirements. Some of these requirements were also considered in the methodology presented in this paper.

Díaz and coworkers [6] describe a seven step methodology that goes from requirement gathering to deployment and maintenance of a WSN for agricultural applications. While the methodology includes several steps, as we do in our methodology, these steps correspond to the life cycle of a WSN application and not specifically to the process of selecting an appropriate solution for the adopted technologies. Furthermore, the paper is focused on a specific application area.

In [12] the authors try to solve the problem of selecting the best technology for a set of requirements in a WSN, presenting a two step methodology. Firstly, the available technologies are reduced to only those that satisfy the gathered requirements. Then, the result obtained in the first step is further refined by taking into account cost requirements. This work considers two use cases to practically demonstrate the methodology and how it helped to solve the deployment of two very different WSNs: one to monitor containers in a port and the other to manage parking lots. Our methodology is more elaborate in the intermediate steps to prune inappropriate solutions.

3 Considered Technologies

For the sake of objectivity and because it is not feasible to consider all the possible technologies available, the focus falls on four different technologies: LoRaWAN, NB-IoT, Sigfox and ZigBee.

This section gives a brief introduction to these chosen technologies and presents a comparative table at the end.

3.1 LPWA Technologies

Under the LPWA umbrella there are three main technologies: NB-IoT, Sigfox and LoRaWAN. Sigfox was the first to appear and is now a mature commercial product. Following Sigfox, the LoRa wireless protocol appeared, which serves as base for the LoRaWAN technology. LoRaWAN has the backing of the industry through the LoRa Alliance. Finally, the most recent technology is NB-IoT, after it became a standard in the 3GPP Release 13. NB-IoT is still in its infancy but, since it is being pushed by Telecom companies, it is expected to become as ubiquitous as 4G or LTE are today.

Sigfox Sigfox is a proprietary technology created by a private company. Provides long range coverage and a small energetic footprint. Transmits over an unlicensed spectrum which makes Sigfox comply with duty cycle, therefore it is subject to noise and interference from other devices that may be transmitting and to the restriction of time it may transmit.

Normally Sigfox partners with other companies to manage the infrastructure. The client is only responsible for the terminal Sigfox modules, the base stations and the Sigfox Cloud are closed to them.

LoRaWAN LoRaWAN is a LPWA communication standard which is maintained by the LoRa Alliance, an open and non-profit association. Like Sigfox, LoRaWAN transmits over an unlicensed frequency therefore it is susceptible to the same shortfalls as Sigfox when it comes to the interference of other devices and the need to comply with the duty cycle [3]. However, LoRaWAN is a fully open protocol (except the physical layer) with access to the full architecture.

LoRaWAN modules are divided in three classes [4, 3]. Class A is the default class and the communication is started at the end node leaving a time frame where it is possible for the gateway to answer. Class B is similar to Class A but has predefined temporal instants where a heartbeat is sent allowing for a downlink transmission. Finally, Class C reduces latency by maintaining the receiver awake when not transmitting.

NB-IoT Narrowband IoT is a standard defined by 3GPP on Release 13, which runs on top of the current GSM/LTE networks. It uses a low frequency licensed spectrum in the 700MHz-900MHz range. These low frequencies allows for more range of coverage and deep indoor penetration. The fact that the frequencies are licensed, frees them from the noise that is common in unlicensed frequencies, improving greatly the quality of service [11].

The reason why it was implemented was to create an alternative technology that had been thought out from the ground up to serve IoT applications, satisfying the requirements that GSM/LTE networks could not. NB-IoT is very similar to LTE. In fact, NB-IoT is LTE with some tweaks (restricting some parts and extending others) making it a better fit to the requirements of IoT applications [7].

3.2 ZigBee

ZigBee is based on the IEEE 808.15.4 specification, which is the standard for Low-Rate Wireless Networks. It predates LPWA technologies and is more mature. ZigBee adds mesh topology support.

The biggest advantage of ZigBee over LPWA technologies is that even though the latter are equipped with several different deployment methods, ZigBee allows for point to point communication. Creating a mesh like topology is not possible in LPWANs, which invalidates them for applications that require the use of such topology, for instance to cover large monitoring areas.

Several parameters influence Zigbee's performance: the distance between nodes, their positioning (if in line of sight or not) and the environment surrounding the devices, such as the atmosphere, traffic between nodes (that can cause interference in the signal), etc. These become more apparent as distance increases between nodes. When the aforementioned concerns are taken into account, ZigBee's limitations surface, showing that it might not provide such a straightforward deployment.

In contrast with ZigBee, ZigBee PRO uses more powerful radios, but also requires more energy, which limits the autonomy of nodes. With the added range we can setup multiple nodes far away from the gateway, allowing for more flexibility in the deployment.

3.3 Comparative table

Table 1 summarizes the characteristics of the technologies briefly described in the previous section. It allows a more direct comparison of their technical differences and similarities.

4 Application requirements

In this section several application requirements will be introduced and described. These requirements were deemed the most relevant out of many more. It should

Table 1. Comparison between LPWAN technologies.

	NB-IoT	LoRaWAN	Sigfox	ZigBee	ZigBee PRO
Frequency	700-900 MHz	868 MHz	868 MHz	2.4 GHz 915 MHz 868 MHz[2]	2.4 GHz 915 MHz 868 MHz[1]
Licensed	Yes	No	No	No	No
Data rate	190-250 kbps[5]	250bps-50 kbps[4]	<100-600 bps[10]	250 kbps	250 kbps (at 2.4 GHz) 10 kbps (at 915 MHz) 100 kbps (at 868 MHz)[1]
Payload	1600 bytes[9]	19-250 bytes 12 bytes overhead[4]	12 bytes[10]	100 bytes (no security) 82 bytes (with security)[2]	variable: 1 octet with payload size[1]
Topology	Star	Star	Star	Star/Mesh/Tree[2]	Star/Mesh/Tree[1]
Coverage	1-8km (urban) 25km (suburban)[5] <35km[11]	2-5km (urban) 15km (suburban)[5] <15km[11]	3-10km (urban) 30-50km (suburban)[5] <15km[11]	100 m	average: <300 m (line of sight) 75-100 m (indoor) <1km for sub GHz channels[1]
Battery life	10 years[7]	>10 years[5]	8-10 years[5]	-	-
Base cost	-	<30€	70€(w/ 1 year subscription)	<30€	<45€
Subscription cost	Yes	No	Yes	No	No

be noted that despite giving several hints to what is important to consider for each requirement, it is not possible to account for all possible scenarios or use cases.

For each requirement, a small list of questions is provided allowing for a better understanding of the specific problems each requirement is associated with.

4.1 Explicit network requirements

Availability The network is composed by several types of components like end-nodes, gateways and central servers, all of which may fail in ways that compromise the ability to communicate. Availability requirements express the need that the network is operational with some probability.

Latency Communication latency refers to the time needed for a message to be sent from a source to a destination, usually expressed in time units (e.g., milliseconds). Real-time applications require a bounded latency.

Packet loss Packet loss may occur due to several reasons, like interference or fading. It has a negative impact on the performance of applications and is defined as a rate between lost packets and total packets sent.

Data Throughput Data throughput characterizes the amount of data that can be, or must be, transferred within a certain time frame, hence being usually expressed as a rate of bits per second (bps). The channel data rate, which is the maximum amount of data that a channel can transfer per unit of time, must be larger than the data throughput required by the application.

4.2 Implicit network requirements from application characteristics

Size The application size, measured in terms of the number of sensor nodes and sensor data produced by each node, implicitly imposes requirements on the network technology to be used. Based on the application size it may be possible to determine throughput requirements for the network.

Autonomy Autonomous or unattended operation of sensor nodes impose constraints on energy consumption. Therefore, WSN applications involving autonomous operation impose an implicit requirement of energy-efficient networking to maximize the lifetime of energy sources.

Autonomy requirements only apply to certain devices in the network. Gateways or coordinator nodes are typically required to be provide continuous service and therefore they are likely externally powered.

Mobility Some applications require network devices to physically move during operation. This translates to a constant mutation in the network topology and to autonomy requirements, which impose constraints to the networking technologies that may be used. As a result, it might not be possible to ensure at deployment time that all nodes remain within a certain distance of other nodes. Thus, upper bounds for distance must be considered when selecting and configuring a network. Mobility also brings forth the problem of dynamic routing which adds further constraints.

Geographical scale WSN applications need to cover a physical area, which may be significantly wide. Knowing the span of the WSN is fundamental for choosing a technology that can provide the required coverage. Depending on the technology, coverage can be solved by using multi-hop or large distance single-hop transmission.

Surrounding environment Characterizes the physical environment surrounding the nodes on the network. It has an effect in the transmission of data. Buildings, interference from other transmitting devices and trees, among many other natural obstacles, influence the reliability of the transmission and hence impose additional requirements on transmission power and transmission distance.

Network scalability Scalability requirements of a WSN application express the needs for seamless addition of new nodes without affecting the network operation. If scalability is needed, then this might affect the decision on the technology to be selected.

4.3 Implicit network requirements from a management perspective

Homogeneity Homogeneity of a network has impacts on its cost and on interoperability. The interoperability provided by an homogeneous network allows for simpler maintenance and management of the infrastructure. Another positive aspect is that personnel only needs to master that specific technology. Naturally, homogeneity imposes constraints on the technology selection.

Cost Many factors can influence cost when choosing a wireless technology for a WSN. The hardware, building and maintaining the infrastructure, training of personnel and service subscription fees, all weight on the final decision. Depending on the context, cost can either be seen as a one-time expenditure or continuous over time.

5 Methodology

5.1 Introduction and Assumptions

The proposed methodology utilizes a graph that conveys the physical network topology. A graph is used because it is the most straightforward way of representing a network. Graph vertices represents nodes and edges represent connections between nodes. To build the graph, we define four distinct steps: establishing the vertices, linking the vertices, superposing partial graphs, and defining the solution search space. The first three steps can be viewed as a graph construction process whose output is employed in the fourth step. These steps will be more thoroughly discussed in Section 5.2.

In order to define the methodology for selecting appropriate technology solutions, the following assumptions about the WSN application and the solution space were made:

- The approximate position where individual devices should be deployed is known.
- The technologies to be considered as possible alternatives is limited to those introduced in Section 3.
- There is at least one central point in the network (gateway) that is the destination of the data flow. Mesh like topologies are not considered. Mobile devices will also not be considered.

5.2 Building the graph

Basic concepts A vertex in the graph represents an individual node on the network and contains the pertinent information about the requirements imposed by that node. Implicit requirements are stored in the vertices.

Edges, on the other hand, represent the physical links between nodes in the network. They contain the explicit requirements that characterize the communication between those two points on the network as well as the difficulties/obstacles (transmission environment) to data transmission.

Establishing the vertices The first step to build the graph is to locate the geographical points of the network and adding them to the graph as vertices. The first set of requirements that are analyzed are implicit requirements. These requirements give a superficial, non-technical and general understanding of the network. Translating this to the graph would mean annotating the vertex with information such as: the amount of data sent and received by the node, its surrounding environment, the level of autonomy needed and its purpose in the network (gateway, end node, relay node).

Although this information is still not enough to rule out candidate technologies, it provides a solid foundation to build the remaining graph.

Linking the vertices The second step consists in establishing edges by connecting vertices. The edges need to be annotated with the information regarding the communication between the two vertices: the maximum expected latency, the minimum throughput required, the maximum accepted packet loss, the physical distance of the link and if there is a line of sight. This process requires the following series of steps:

1. Select a vertex not previously processed.
2. Connect the vertex to all other vertices.
3. Analyze one edge at the time, add the communication requirements imposed by the two connected vertices.
4. Consider the available technologies, remove the edges whose requirements cannot be met.
5. Iterate over the next set of vertices taking into account the inheritance of requirements.
6. Once all the vertices of this partial graph have been dealt with, jump to step 1 to generate another partial graph.

The link requirements on the edges are associated with the vertices. The information stored in the edges answers the explicit requirements of the application with a finer granularity, which allows decision making for the viable technologies to serve each particular edge. Therefore, the two nodes that are connected and their respective requirements become more relevant than if we did not have this detailed view of the network.

Not only does this approach allows for micro decision making between two nodes, but it also brings forth the notion of inherited requirements. Inherited requirements shine light on a problem that might go unnoticed and cause problems to the WSN if not caught at this early stage. As an example, consider three vertices A, B and C, edge 1 (connecting A to B) and edge 2 (connecting B to C). If vertex A requires sending large amounts of data, then edge 1 would have the requirements necessary to reliably transmit that data. In turn, both vertices B and C have a smaller requirement sending less data, therefore edge 2 would have a weaker requirement than edge 1. This could possibly degrade the transmission quality of service.

It is important that the information on the vertices and edges be detailed and concrete. If detailed information cannot be given, then at least an estimation of the requirements should be provided. A solution for this would be categorizing requirements into a scale, according to the characteristics of the considered technologies.

Superposing graphs The result of the process described in Section 5.2 is a set of partial graphs that can either have the same or different paths to the sink.

The graphs are superposed into a single layer, thereby creating a single graph. Due to the possibly large number of graphs created in the previous step, it is normal that in the resulting graph there are edges that connect the same vertices but with different requirements. Excessive requirements could have an impact on the complexity of the next step, thus pruning is necessary. The pruning procedure is as follows:

1. If the edge only has one requirement, then that requirement holds.
2. If an edge has several equal requirements from different paths where inheritance has not appeared, that means the previous path does not matter, therefore just one set of requirements is needed for the path.
   (a) Special case: requirements that are equal but due to inheritance need to remain.
   (b) Redo step 2. but for inherited requirements, if there are more than one.

Another effect of having a common edge with multiple requirements is that it can help mitigate an overzealous inheritance of requirements that could lead to edges having stricter requirements than they would need. This over inheritance is dependent on the starting vertex and its requirements as well as the paths followed. If other graphs were not considered, then this problem could not be contained.

At this point, new functional requirements for the network are specified. The practical consequence of these new functional requirements is that some paths will now have constraints. The cuts in edges done in the previous step ensure that the functional requirements and the paths that they impose are possible.

5.3 Definition of the solution set

Having a graph representation of a network is not by itself a methodology. This section proposes a process to interpret any received graph built as specified in Section 5.2 and to reliably generate a set of possible solutions.

The edges of the superposed graph can have more than one set of requirements depending on the previous path taken up to that edge. Simultaneously, for each edge there is also at least one or more technologies available. This creates two questions: (1) which set of requirements do we choose to cover? and (2) which technology do we choose for each edge?

These questions are not directly answered but are dependent on the final solution chosen. This solution also gives the complete topology of the network. To achieve this final solution space an algorithm is used:

1. Pick a vertex not previously chosen.
2. Pick an outwards edge from the previously chosen vertex.
3. Of the available set of requirements, pick one not previously picked.
4. Of the available set of technologies, pick one not previously picked.

5. Repeat from step 2. until a sink is reached.
6. Add the resulting path to the set of paths starting in the vertex chosen in step 1.
7. Repeat from step 2. until all possible combinations of requirements and available technologies from all edges have been exhausted.
8. Go to step 1 until all starting vertex have been subject to the algorithm.

The output of this algorithm is all the possible combination of paths from each vertex to the sink. In other words, for each vertex there is a corresponding set of paths starting on that vertex and ending on the sink vertex. The problem is that the solution space is massive and not ordered by any sort of metric that would allow for an objective choice to be taken.

Graphs can be rated by using criteria other than the requirements already used when building the graph. The rating of these criteria serves as input to an algorithm to compute the Pareto front of solutions that have the best compromises between the criteria. It is the network designer that finally has to pick one of the solutions that exhibits a desirable or acceptable compromise among the criteria

The procedure is defined as follows:

1. Pick a starting vertex (not previously chosen) and its corresponding set of paths.
2. For each path on the set:
   (a) Choose a path from each of the other starting vertex sets. This results in a solution graph.
   (b) Rate the graph according to the defined criteria. The result is a tuple of values.
   (c) Run the tuple of values through the Pareto front algorithm.
   (d) If not removed by the previous step, add the tuple to the current set of solutions.
   (e) Return to step 2.(a) until all paths of all other starting vertex sets have been used.
3. Go to 1.

Rating a graph is subjective and depends on the criteria in question. Even then, there could be various ways of rating a graph for the same criteria. Homogeneity could be rated by the number of different technologies used in the graph and cost by the accumulated cost of each edge on the graph (this would exclude managerial cost of the network).

6 Use Case - Seixal Bay

The use case is related to the AQUAMON project which has as one of goals the deployment a WSN. This WSN will be deployed in the Seixal Bay (southern bank of the Tagus river) with the purpose of taking water quality measurements and to study various aspects of tidal movements. Initially it will consist in five distinct

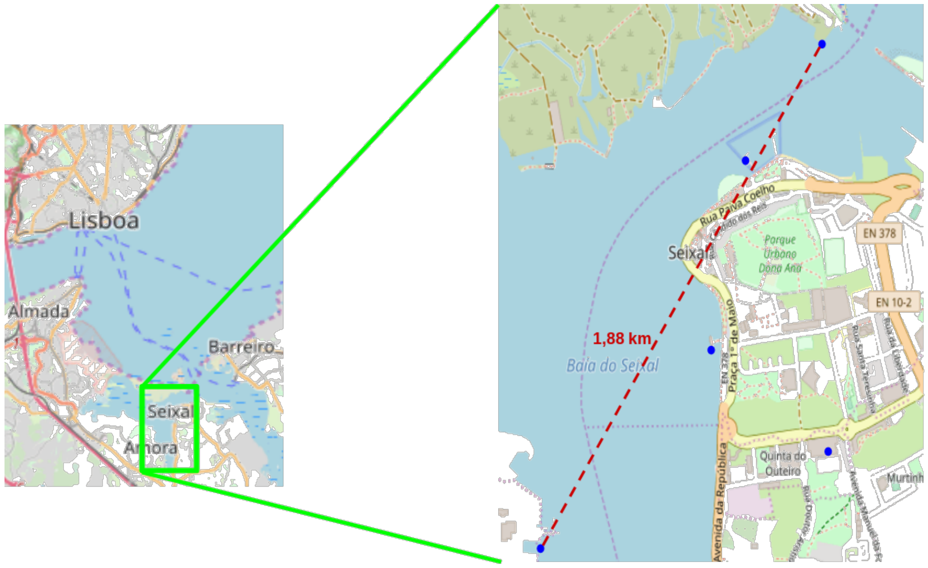

Fig. 1. Map overview of the deployment area - Seixal Bay.

sensors/nodes in fixed positions along the waterside. One of the objectives is to provide easy and open access to the data gathered.

The characteristics of the site provides a great environment to test the methodology. As can be seen in Figure 1, the area covered by the WSN contains all sorts of environments: from open waters to dense urban environments, nodes with and without line of sight between each other, some at large distances from others (the largest distance between two nodes in the network is approximately 1,88km), some closer. This difference in environments, that change in just dozens of meters, is a true test to the feasibility of the methodology.

7 Conclusion

This paper presented a solution to the problem that is choosing a wireless technology responsible for the transmission of data in WSNs that satisfies application and functional requirements. In addition, our solution also mitigates exterior, non-technical influences such as human bias from the choosing process. Our automated process comes in a form of a four step methodology and introduces the novelty of using graphs as a way of translating, analyzing and outputting a solution in such a way that, in its final stage, allows the network designer to choose in which criteria to base his final solution in.

The methodology presented in this paper constitutes a solid foundation for future work.

8 Acknowledgements

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) under LASIGE Strategic Project UID/CEC/00408/2019 and AQUAMON project (PTDC/CCI-COM/30142/2017).

References

1. ZigBee PRO specification
2. ZigBee Specification
3. Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., Watteyne, T.: Understanding the limits of lorawan. IEEE Communications magazine **55**(9), 34–40 (2017)
4. Alliance, L.: Lorawan a technical overview of lora and lorawan. Tech. rep., Huawei
5. Chen, J., Hu, K., Wang, Q., Sun, Y., Shi, Z., He, S.: Narrowband internet of things: Implementations and applications. IEEE Internet of Things Journal **4**(6), 2309–2314 (2017)
6. Díaz, S.E., Pérez, J.C., Mateos, A.C., Marinescu, M.C., Guerra, B.B.: A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. Computers and electronics in agriculture **76**(2), 252–265 (2011)
7. Finnegan, J., Brown, S.: A comparative survey of lpwa networking. arXiv preprint arXiv:1802.04222 (2018)
8. Mac Ruairí, R., Keane, M.T., Coleman, G.: A wireless sensor network application requirements taxonomy. In: 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008). pp. 209–216. IEEE (2008)
9. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: A comparative study of lpwan technologies for large-scale iot deployment. ICT Express (2018)
10. Sigfox: Sigfox whitepaper. Tech. rep., Sigfox (2017)
11. Sinha, R.S., Wei, Y., Hwang, S.H.: A survey on lpwa technology: Lora and nb-iot. Ict Express **3**(1), 14–21 (2017)
12. Vannieuwenborg, F., Verbrugge, S., Colle, D.: Choosing iot-connectivity? a guiding methodology based on functional characteristics and economic considerations. Transactions on Emerging Telecommunications Technologies **29**(5), e3308 (2018)