Dynamic autonomous set-up of relays in Bluetooth mesh

Javier Silvestre-Blanes
Dept. Comp. Eng. (DISCA)(ITI)
Universitat Politècnica de València
Alcoy, Spain
ORCID: 0000-0001-7091-0040

Juan Carlos Garcia Ortiz
Instituto Tecnológico de Informática (ITI)
Valencia, Spain
ORCID: 0000-0003-1920-0961

Víctor M. Sempere-Payá
Dept. Comms (DCOM)(ITI)
Universitat Politècnica de València
Alcoy, Spain
ORCID: 0000-0001-9271-2010

***Abstract*— BLE-based mesh networks are based on a simple flooding algorithm with some mechanisms to reduce network saturation, called managed flooding. The operating parameters of the network establish its performance, but in an industrial environment the operating conditions are not permanent, so a system that can adjust to these changes is necessary. A global decision system is not valid since each part of the network may have different properties. An autonomous system that does not introduce an overhead of message exchange is necessary for its operation. This paper proposes an algorithm based on the information provided by a single control message exchange that allows each node to autonomously select its operating parameters to improve the quality of links with neighbouring nodes and thus improve the overall performance of the network.**

Keywords—mesh networks, Bluetooth, BLE.

I. Introduction and Related Work

In today's industry communications networks are, from the point of view of the technologies used, increasingly heterogeneous, with wireless communication taking up an ever-greater role due to the flexibility they provide [1]. There is a wide variety of wireless technologies (BLE, WiFi, Zigbee LoRaWAN, NFC, 3g/4g/5g, etc.), but no single solution that can be coupled to different requirements. [2][3]. In Mesh topologies, nodes can be interconnected directly and dynamically, without a hierarchy, allowing many-to-many communications to efficiently carry data from source to destination. Then, Mesh networks can play an important role in this field, both in device-2-device (**D2D**) and Wireless sensor network (**WSN**) communications. The use of mesh networks based on the **BLE** (Bluetooth Low Energy) protocol has significant potential [4]. These have been used in precision agriculture applications [5], for smart home [6] and buildings [7] applications, by robots in logistics [8], in industrial control applications [9], for real-time applications [10] or in industry 4.0 [11].

In these networks there is no routing. Instead the standard provides an efficient flooding algorithm to reduce the saturation that this mechanism generates. The nodes of these networks are configured with a transmission power, as well as other message retransmission parameters that determine the range and quality of the links. In [12] an algorithm is presented to determine the transmission power, data rate and connection interval in **BM** (Bluetooth Mesh) using only a connection schedule (not connectionless) and with direct connection (star topology) between master and slave. The trade-off between energy efficiency, reliability and throughput for each PHY mode of BLE is analysed in [8], but in master/slave communications and for connection-based mode. A retransmission threshold determination method based on statistical analysis of information loss is presented in [9]. The proposed method has been implemented as a data-collection protocol for the central network node in a star topology. The system is based on a previous transmission of messages. In [10] the benefits of the flooding protocol are highlighted, although it is pointed out that it can be inefficient in one-to-one communications in large networks, especially from an energy point of view. In this work the *Fruitmesh* protocol is introduced, which requires the introduction of additional messages (clusterwelcome, joinme, etc.). In [11] it is proposed to perform, in real time and in a distributed manner, a tuning of the parameters of retransmissions and transmission power, for which a considerable exchange of additional messages is also necessary. In connection-oriented systems, from Bluetooth 5.2 onwards, there is the LE Power Control protocol that allows to dynamically adjust, depending on the RSSI of the packets, the transmission power between the master and the slave [17].

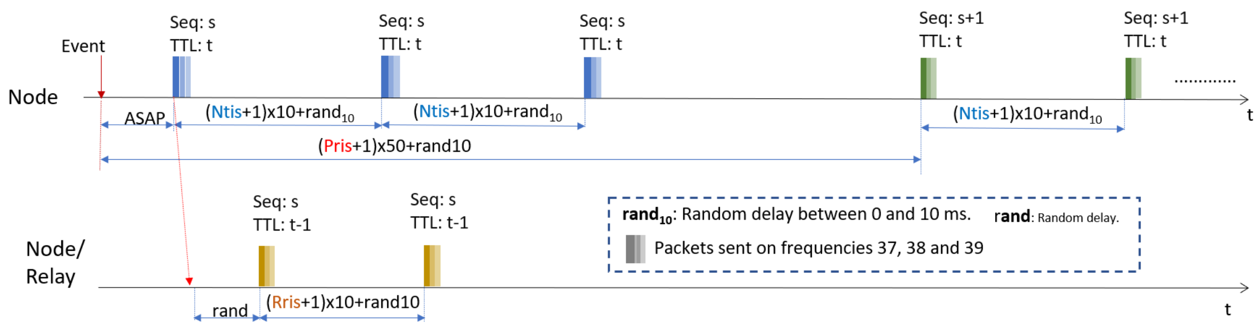

Fig. 1 Timing of message sending

In this work, a mechanism for mesh networks is proposed to determine the operating parameters of the relays, in a connectionless communication based on advertising bearer, based solely on the exchange of a message whose periodicity can be adjusted according to the trade-off between response speed and network saturation plus energy consumption.
Section II briefly describes the BLE mesh and the operational parameters that influence network performance, Section III presents the algorithm. Section IV presents and analyses the results obtained and finally, in section V, conclusions and future work are presented.

II. BM and Protocol Proposed

Bluetooth has been one of the most common protocols in the field of wireless personal area networks for many years. With the successive improvements in the different versions of speed, security and robustness, its use has been extended to other fields, especially in the field of IoT and industry. This happened especially from the version called BLE (Bluetooth Low Energy), making possible its use in scenarios with restrictions in energy consumption. It is not the purpose of this paper to make a detailed description of the protocol, and there are excellent descriptions such as [18][19], although it is necessary to describe minimally some elements that allow the text to be fairly self-contained.
BLE operates in the 2.4 GHz band (Industrial Scientific Medical), using a Frequency Hopping-Spread-Spectrum mechanism to provide robustness and reliability in communications. The 40 channels it can work with are divided into three primary channels (37, 38 and 39), in different non-adjacent segments to minimize the effects of interference with the Wifi protocol, and 37 secondary channels covering the rest of the spectrum. Notifications, node discovery and scanning process, and in general all network maintenance messages are sent through the primary channels.
The nodes in a BM network can be clients, servers, relays, as well as others such as friend node, proxy node, low power node, or provisioner node. These last nodes have the mission to connect the BM network with nodes that are not part of it, to provide communications to nodes with very low power consumption that have their radio off, etc. But they are not the subject of this paper.
In the BM network, the information will flow from the server to the client, directly, or through the Relay nodes that rebroadcast the information they receive over the network. The objective of these retransmissions is to improve the reliability offered by the network, which can be affected by different common problems in wireless communications [20], such as collisions or multipath propagation among others. In addition, the client and server can in turn be server and client with other nodes, and also act as relay nodes.
There are three different types of physical level, LE_1M, LE_CODEC and LE_2M, with different data rate or range properties. The paper will assume a physical level of LE_1M, which is the one introduced by standard 4.0, and which is the one used by BM.
The routing protocol in BM is based on managed flooding, which makes routing algorithms and exchange of route management messages unnecessary. Each message received by a node is forwarded by the node without prior knowledge of the number and status of neighbouring nodes. In order to

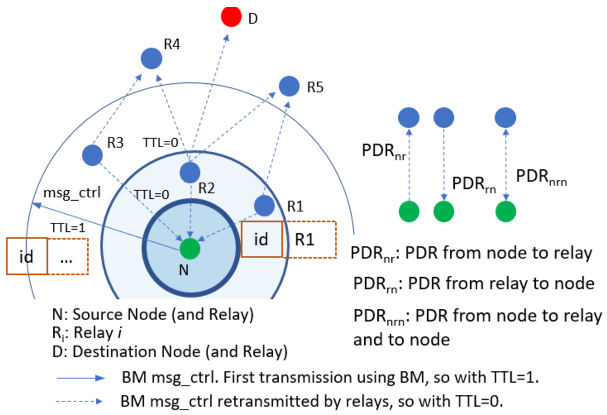

Fig. 2 Transmission of a message control

avoid or reduce the saturation of messages in the network, the protocol has the following optimisations:

- Relay cache: in which each node of type Relay maintains a list of packets already received/diffused. If a new received packet is found in this list, is immediately ruled out.
- Time To Live (TTL): where each packet has a limited maximum number of retransmissions. Once this number is reached, no further retransmissions of this packet are made.

In addition, it is possible to establish whether or not the nodes can act as relays.
The parameters related to retransmissions are:

- Network Transmit Count (NTC) and Network Transmit Interval Steps (NTIS): which indicates the number of retransmissions of a packet and the time instants in which they are made. A packet is broadcast NTC + 1 times, i.e., an NTC value of 0 indicates that the message is only transmitted once. On the other hand, the repetition interval in milliseconds is set as *(1 + NTIS) x 10*, the default NTIS value is 1.
- Relay Retransmit Count (RRC) and Relay Retransmit Interval Steps (RRIS): which indicates the number of relay retransmissions of a packet and the time instants in which they are performed. A message is retransmitted RRC + 1 times, i.e., an RRC value of 0 indicates that the packet is only retransmitted once. On the other hand, the repetition interval in milliseconds is set as *(1 + RRIS) x 10*, the default RRIS value is 0.
- Publish Retransmit Count (PRC) and Publish Retransmit Interval Steps (PRIS): which indicates how many times an application-level message is retransmitted and the time instants in which they are performed. A message is retransmitted PRC+1 times. The repetition interval in milliseconds is set as *(1+PRIS) x 50*, the default PRIS value is 1.

The effect of these parameters can be seen in Fig. 1, using (NTC=2; RRC=1; PRC=1). When an event occurs that involves the transmission of a packet, it is generated as fast as possible (ASAP) and transmitted on channels 37, 38 and 39. This is retransmitted three times at network level (in blue in the figure), since NTC=2. Nodes close to the sender acting as relays, when receiving the message and if the received TTL is greater than 0, will decrement it and retransmit the message. In addition, they will do it twice if RRC=1. The other network nodes within range of the transmitting and relay nodes, if they have also correctly received the first message, will receive, and discard the other messages sent by the node, and the two sent by the relay upon detection of the message in the relay cache.

The other parameter to be negotiated is the TxPower. Specifically, nodes can transmit with the following powers (in dBm): -20, -16, -8, -4, 0, 4, 8, 16, 20 (not all powers are supported by all implementations) which affects the range of the message, and the energy consumption of the node. Although the differences in consumption between different TxPower are only non-negligible from 0dBm onwards, increasing the power can improve the range and the signal received at the receiver, and this can cause it to reach more relays, thus increasing the retransmission overhead. Finally, the number of relays in service could also vary and affect the network performance.

III. ALGORITHM

The system is based on the transmission of a control message with a pre-established periodicity in the system, which allows us to regulate between response speed and energy consumption/network saturation needs. Each node contains a control table structure. Each node will transmit a *msg_ctrl* with *TTL*=1 and with a known periodicity (called T_{ctrl}). Only retransmissions at the network level are considered, since retransmissions at the application level are only possible on nodes, but not on relays. This message will reach the relays within the coverage area (R1, R2 & R3 in Fig. 2), which will retransmit this message and therefore their responses will be received by the transmitting node (N in Fig. 2). Other relays further away from the sender and which have not received the original message, will receive the message from the relays, but will not retransmit it as they have TTL=0, or will discard it if it arrives in duplicate (R4 and R5 in Fig. 2). The sender node of the control packet notes in the control table the generation of the message and a unique identifier. This information is not altered during retransmissions. In this way, the sender node, on receiving the retransmission from a relay and given that this message includes the address of the relay, knows which relay it comes from (In fig. 2, only the response of R1 is shown). Through this exchange, the source node will be able to calculate the *PDR* (*Packet Delivery Ratio*) of the node-relay[i]-node communications, called $PDR(i)_{nrn}$. The different relays also have the *msg_ctrl* sending mechanism activated. Therefore, the node knows the number of messages it should have received with respect to those actually received in a given period of time, that is, $PDR(i)_{rn}$. It can thus calculate the *PDR* from the Relay to the node, $PDR(i)_{rn}$. In this calculation, the node does not know the *RRC* value that relay is using, as the algorithm can locally modify this value at each node, but it can evaluate the quality it perceives of the messages sent by the Relay, regardless of that value. Therefore, it can calculate the PDR from the node to Relay *i*, as follows:

$$PDR(i)_{nr} = \frac{PDR(i)_{nrn}}{PDR(i)_{rn}} \quad (1)$$

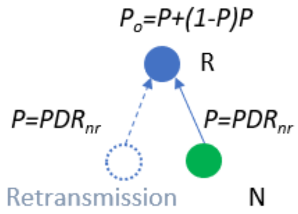

Fig. 3 Influence of retransmissions on the PDR

TABLE 1 Quality vector values

Quality vectors					
QR (RRC)	0	1	2	3	
QT (dBm)	8	4	0	-4	8

Algorithm 1: update params

Input: NN; LC

```
NR=getWindowsReliability()                  #equation (1)
stability = WindowsStability (NR,NN)        #Alg. 2
decision = fun_decision (stability, NR, NN) #Alg. 3
CC[0]=NONE; CC[1]=NONE
if (decision==PARAMS_UP) then
  if ((LC[0]==NONE) OR (LC[0]==UP & LC[1]=TX)) then
        qr++, CC[0]=UP; CC[1]=REP
  else qr--; qt++; CC[0]=UP; CC[1]=REP
else if (decision==PARAMS_DOWN) then
  if ((LC[0]==NONE) OR (LC[0]==DOWN & LC[1]=TX)) then
        qr++, CC[0]=DOWN; CC[1]=REP
   else qr++; tx--; CC[0]=DOWN; CC[1]=TX
LC=CC
```

The value of PDR_{nr} being the average value of the above. This value is considered the node reliability of the node (***NR*** in the algorithms, where ***NN*** is the number of nodes).

Regarding the increase in the number of retransmissions, this will affect the *PDR* observed at the destination, in the same way as if an additional Relay were added per number of retransmissions, with the particularity that in this case there would never be interference in the transmissions. Fig. 3 shows the influence on the *PDR* observed by the receiver (*Po*), assuming that a retransmission is equivalent to the retransmission of another virtual Relay. Thus, assuming that the distance, power and environmental conditions give us a PDR_{nr} from the Relay to the destination (whether this is another Relay or the final destination), called *P* for simplicity, at the destination the observed *PDR* will be $P_o=P+(1-P)P$. If another relay were to be added, the new *PDR* that would be perceived would be $P_o+(1-P_o)P$, and so on.

In addition to the above-mentioned control table at each node, there are vectors *QR* and *QT* that contain the *RRC* and *TxPower* values that can be used (See Table 1), *qr* and *qt* being the indices of the values used in the algorithm. The nodes, besides sending the *msg_ctrl* with the established periodicity T_{ctrl}, in each reception of *msg_ctrl* the values of number of neighbor nodes (*NN*) and the information of the tables are updated. The limits of reliability, minimum (*MIN.*), maximum (*MAX.*) and what is considered to be stable reliability (*STABLE*), within a tolerance (*TOL.*) in order to avoid continuous changes. Each calculation window (*W*),

Algorithm 2: WindowStability(NR, NN)

```
Input:
    if (NR<0) then
        stability=-1
    else if (NR < STABLE) then
            stability = max( (NR-STABLE)/(STABLE-MIN); -1)
        else stability = min( (NR-STABLE)/(MAX-STABLE); 1)
    return stability
```

Algorithm 3: fun_decision(stability, NR, NN)

```
Input:
1   if (abs(stability) > TOL) then
2     if (stability<0) decision=PARAMS_UP then
3     else decision=PARAMS_DOWN
4   if (NN <= MIN_NODES) then
5         decision=PARAMS_UP
6     else if (NN>=MAX_NODES) then
7         decision=PARAMS_DOWN
```

composed of several T_{ctrl}, will call the ***update_params*** algorithm (alg. 1). This will call the ***WindowsStability*** function (alg.2), which calculates the normalized stability value for the obained *NR* value and the above mentioned values. This value will be closer to 0 when we are closer to *STABILITY*, and will tend towards -1 when we are closer to *MIN* and to 1 when we are closer to *MAX*.

Fig. 4 shows the result of the calculation of the normalized stability value [1,1] according to the Reliability values and the configuration parameters conf0 and conf1 used in the following section and shown in Table 2. Once this value is known, the ***fun_decision*** (alg. 3) function will be called to establish whether to raise or lower the operating parameters. When the normalized value of stability is below *-TOL*, it will be set to raise the quality parameters, while if it is above *+TOL*, it will be set to lower them (lines 1, 2 and 3, Alg. 3). Another value that is considered within this function is the value of NN. If the number of neighbors is below the MIN_NODES threshold, then the node will raise its transmission parameters. On the other hand, if NN is above an upper threshold, called MAX_NODES, then it is considered that there are many alternatives of neighbors that can retransmit the information and the quality parameters will be lowered.

After the calls to the two functions, the naive algorithm used modifies the quality parameters in a non-aggressive way and alternates between the two parameters being handled (lines 4-13, Alg. 1). Since it is a mesh network, the nodes have no

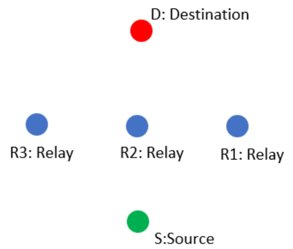

Fig. 5 Scenario

information about the network topology. The objective of the algorithm is to provide good reliability without consuming excessive resources. Therefore, when the reliability is low, transmission parameters will be increased, but if the reliability is very high, the transmission values will also be reduced, as long as it has a certain number of neighbors. The algorithm has information about the changes made in the previous *W* window, in the *LastChange* (*LC*) vector, which contains in *LC[0]* the values *NONE*, *UP* or *DOWN*, and in *LC[1]=TX* if *TxPower* has been changed or *LC[1]=REP* if *RRC* has been changed, being the *CurrentChange* (*CC*) vector the equivalent element of the current window. If it is indicated to go up (lines 5-8, alg. 1), in the case that there has been no change in the previous window or *TX* has been raised, the value of the repetitions is raised, i.e. the *qr* index is increased and *CC* is updated. Otherwise, the index *qr* is decreased and the power index *qt* is increased, updating *CC*. If it is indicated that they are to be reduced (lines 9-12, alg. 1), in the case that no changes have occurred in the previous window, or these are to lower the power, the index *qr* is lowered or in the opposite case *qr* is raised and the index of the power *qt* is lowered, updating in both cases the value of *CC*.

TABLE 2 Configuration values

Configuration	MIN.	STABLE	MAX.	TOL.
conf0	0.4	0.7	0.95	0.5
conf1	0.7	0.8	0.99	0.3

IV. TESTBED AND RESULTS

A. scenario description

The scenario chosen to evaluate the performance of the algorithm can be seen in Fig. 5, where all nodes have active

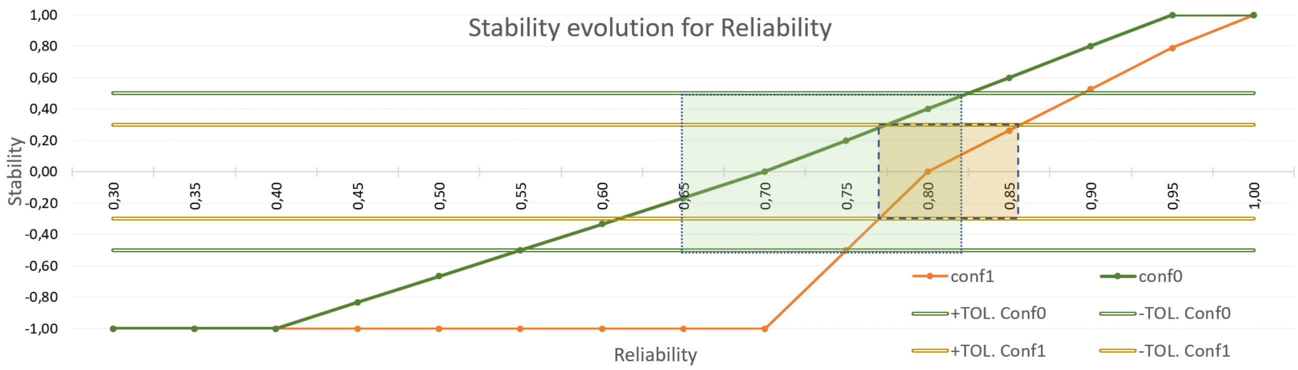

Fig. 4 Reliability and Stability

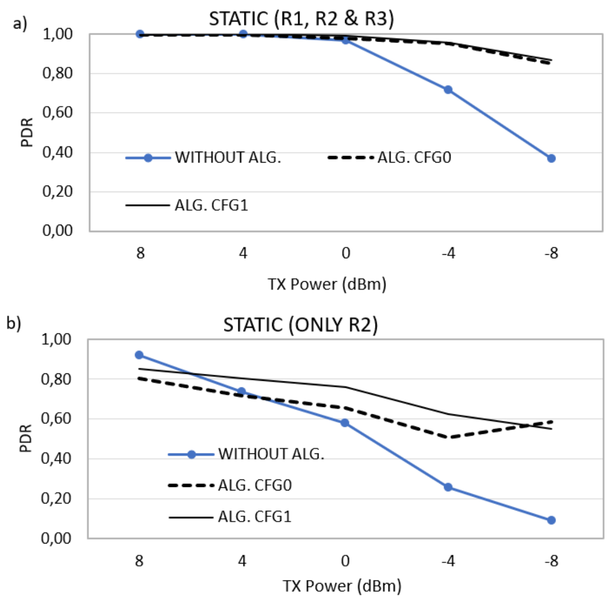

Fig. 6 PDR in static scenarios

relay behavior. BM performance is evaluated through simulations developed in a proprietary software developed in Python (as in [21]). This simulator implements the basic operation of the transport, network and bearer layers (relay control, TTL, timers, configuration values, etc.) allowing us to keep track of the packets that are sent and generated through the simulation. To simulate the range of the signal it is specified through a delivery probability value to the neighboring nodes according to their distance from the sender. The values of the simulator were verified in real implementations at [22]. With an approximate distance between nodes of 30 meters. A *RRC*=0 (qr=0), the TxPower of 8, 4, 0, -4 and -8, and the values chosen for the reliability values of two configurations, *conf0* and *conf1*, as shown in Table 2, are used. These values have been chosen to analyze the performance of the algorithm with two different tolerances to the stability values. Thus, conf0 has a wide margin to consider the realibility acceptable and not to act on the parameters, while with conf1, this margin is narrower and the parameters will be acted before, as shown in Fig. 4. Also a T_{ctrl}=10s and a *W* of 10 are used. Regarding the defined scenario, we will refer of a static scenario when all nodes are connected during the entire test period, and of a dynamic scenario when a node is disconnected during the test period. Specifically, in the second 400, R1 and R3 are switched off.

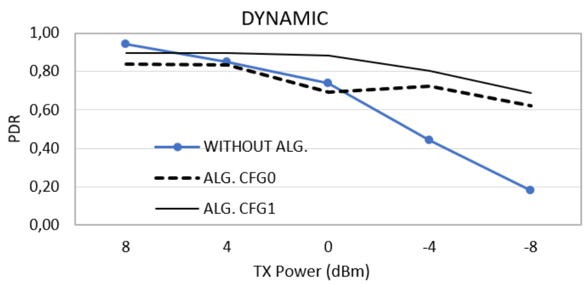

Fig. 7 PDR in dynamic scenarios

B. Results

In the first test the static scenario is considered, and Fig. 6 shows the behavior of the *PDR* obtained in the source/destination transmissions using only one relay (R2) or with all three, for the initial values of *RRC*=0 and TxPower indicated on the x-axis, values that do not change when no algorithm is used. As can be seen, with the chosen working distance, a transmission of 8 dBm provides a high *PDR*, so the algorithms have no room to improve the performance. The use of two more relays increases the paths to the destination, which also improves the *PDR*, so that even with a transmission of 0 dBm, the *PDR* obtained is very high. For TxPower values of -4 dBm (qt=3), the algorithm adapts the parameters to obtain a *PDR* much higher than the one obtained without the use of the algorithm, especially with *conf1*, since it has stricter values. However, this could also be achieved with a correct setup at the beginning, but what would happen in case of changes in the environment? Fig. 7 shows the global behavior in a scenario where in the second 400 relays R1 and R3 are switched off. The algorithm adapts the parameters to suit the reduction in the numbers of relays appropriately, especially for *conf1* for an initial TxPower of less than 8 dBm. For 8 dBm the fixed setting gets a better values because the algorithm will detect that we exceeded TOL. above, and lowers the parameters. In case of not being aggressive in saving the quality parameters and not asking for a parameter reduction when the quality is very high as mentioned in the algorithm description, it would also outperform the operation with the parameters (8 dBm, RRC=0).

Fig. 8 and Fig. 9 show the evolution of the parameters for the two configurations in the source, R2 and destination nodes, where the minimum and maximum values of stability are indicated by the green double line, and for initial values of RRC=1 and TxPower=0dBm. The first ordinate axis shows the values of TX Power, RRC or NN, since all are within the range [-4,+4]. The second ordinate axis shows the stability values, which are in the range [-1.00,+1.00]. On the ordinate axis the calculation window is represented for the specified values of T_{ctrl}=10s and a *W*=10. When the stability is out of range and therefore the parameters have been changed, this is indicated by a red marker. These values are adjusted among the 5 nodes until the second 400, where R1 and R3 are turned off (*NN* goes from 4 to 2 for R2), which causes changes not only in the relay, but also in Source and Destination, to adapt to the new configurations with fewer relays. The values chosen for *conf1* determine narrow stability values, so that small variations cause changes in the parameters.

The algorithm allows us to dynamically reconfigure the operating parameters of the relays in a BM, to maintain a *PDR* between the values determined, where this is possible, introducing little message overhead, and with a reaction speed dependent on *W* and T_{ctrl}.

C. Energy considerations

When evaluating the energy costs of the algorithm, it is necessary to take into account not only the extra cost it adds, but also the savings it can allow in data transmission. Thus, assuming that in a fixed configuration, in the aforementioned scenario, and with t_{ctrl}=10s and W=10, a Tx Power of 8 dBm and RRC=0 are used to guarantee a high *PDR*, the energy consumption is compared with a scenario where the

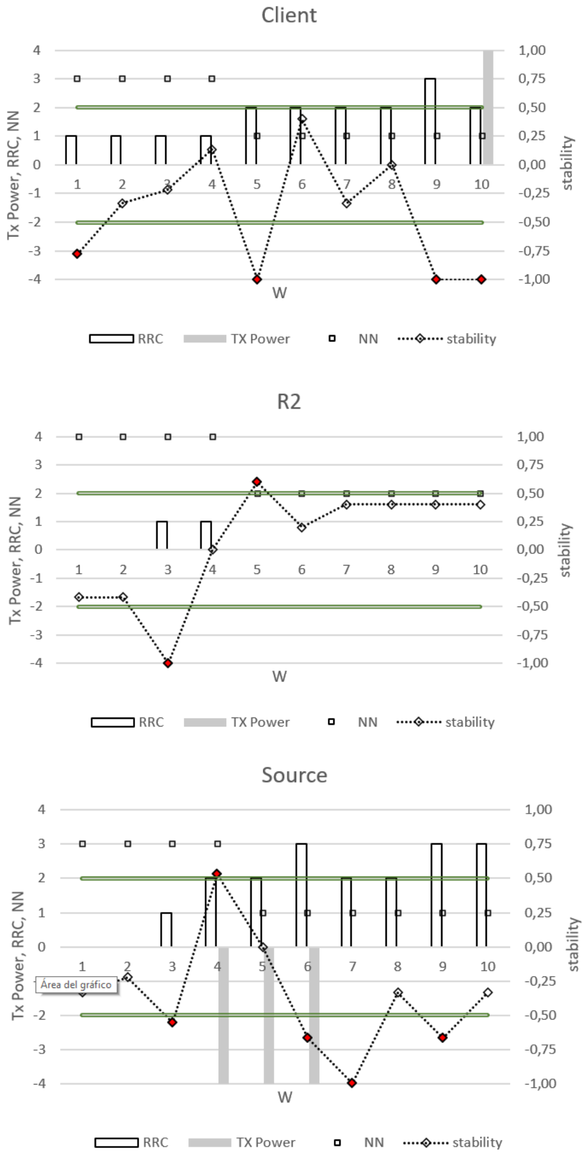

Fig. 8 conf0 evolution

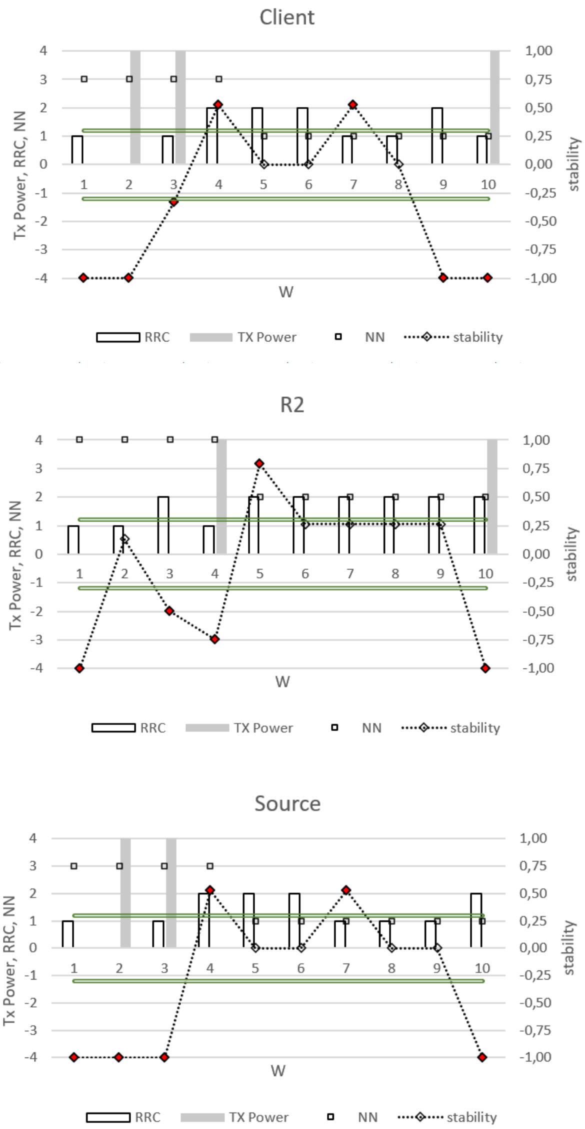

Fig. 9 conf1 evolution

algorithm is used, based on measurements taken on an actual testbed at [22]. In Fig. 10, the energy savings using the algorithm versus the fixed configuration are presented. When the data transmission frequency is very low (1 message per

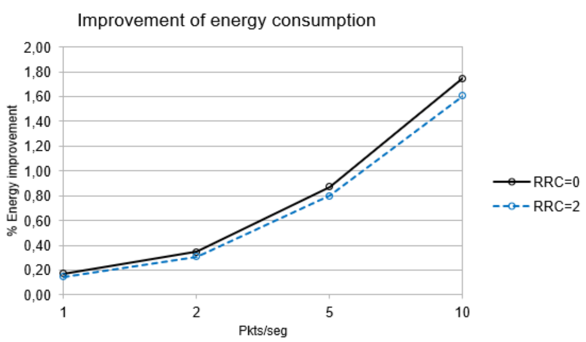

Fig. 10 Energy consumption improvement

second), the energy consumption of the communications is still improved. When the data frequency is higher, such us 10 messages per second, this improvement can represent a reduction of about 2% of the energy consumption. Therefore, the use of the algorithm compared to a fixed configuration does not represent a higher consumption, despite the fact that it involves the transmission of an additional message, but can even save this consumption. This will obviously depend on the t_{ctrl} and W parameters that will affect not only the reliability and speed of adaptation, but also whether it is slightly advantageous or burdensome for the system in terms of energy consumption.

V. Conclusions and Future Work

Mesh networks have great potential for use in the field of industrial communications and IoT, given their many-to-many connectivity, ease of deployment and reliability. The communications configuration parameters have to achieve a compromise between reliability and power consumption, and this in an environment where conditions can vary frequently.

In connectionless networks, there are no methods in the standard that allow this adjustment. The paper proposes a method to perform this adjustment, which also takes into account not only the power of the signal, but also depends on the number of neighbors and the actual and measured reliability in sending packets. Depending on the configuration parameters, it is possible to achieve more or less fast responses to changes, improving connectivity, and without introducing a high message exchange that involves a significant increase in energy costs. Moreover, by allowing this adjustment, at the minimum power required, the savings achieved in data transmission offset the exchange of control messages. The different needs in terms of speed of response and energy consumption are the keys to the adjustment of the T_{ctrl} and W parameters. As future work, we are working on a real implementation, as well as on the development of the simulator to evaluate the behavior in more complex networks with a larger number of nodes, including mobile nodes.

ACKNOWLEDGMENT

This work was supported by Grant PID2021-123168NB-I00 funded by MCIN/AEI/10.13039/501100011033 and ERDF A way of making Europe. It has also been funded by the i4Q project that has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement No. 958205.

REFERENCES

[1] Stefano Scancio, Lukasz Wisniewski, Piotr Gaj. "Heterogeneous and dependable networks in Industry – A Survery", Computers in Industry, vol. 125, p 103388, 2021.

[2] W.Z. Khan, M.H. Rehman, H.M. Zangoti, M.K. Afzal, N. Armi, K. Salah. "Industrial internet of things: Recent advances, enabling technologies and open challenges", Computers and Electrical Engineering 81 (2020), 106522

[3] Miljan Sikimic et al. "An Overview of Wireless Technologies for IoT Network" 19th International Symposium INFOTEH-JAHORINA, 18-20 March 2020

[4] T.C. Yang Lam, S.S. Ling Yew, S.L. Keoh. "Bluetooth Mesh Networking: An Enabler of Smart Factory Connectivity and Management". 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–6.

[5] Luan H.S. Alves, Elida Antunes, Ricardo Ferreira, Jos´e A. M. Nacif. "A Mesh Sensor Network based on Bluetooth: Comparing Topologies to Crop Monitoring," in IX Simposio Brasileiro de Engenharia de Sistemas Computacionais, November, 2019.

[6] Stine Akredalen, "Automatic dynamic tuning of Bluettooth mesh networks for optimal performance". Master's thesis in Cybernetics and Robotics. Norwegian University of Science and Technology. Nordic Semiconductor. July 2022.

[7] Ilaria Sergi, Teodoro Montanaro, Maria C. Gammariello, Luigi Patrono. "The use of Bluetooth Mesh Networking in IoT-aware Applications" 2021 6th IEEE International Conference on Smart and Sustainable Technologies (SpliTech), Bol & Split, Croatia, 2021

[8] Adnan Aijaz. "Infrastructure-less wireless connectivity for mobile robotic systems in logistics: why Bluetooth mesh networking is important," in 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September, 2021.

[9] N. Xia, C. Hsiao-Hwa, C.S. Tang. "Emerging Technologies for Machinetype communication networks", IEEE Network, vol. January/February, 2020

[10] Luca Leonardi, Gaetano Patti, Lucia Lo Bello. "Multihop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks" IEEE Access vol. 6, 2018.

[11] Celia Garrido-Hidalgo, Diego Hortelano, Luis Roda-Sanchez, Teresa Olivares, M. Carmen Ruiz, Vicente Lopez. "IoT Heterogeneous Mesh Network Deployment for Human-in-the-Loop Challenges Towards aSocial and Sustainable Industry 4.0" IEEE Access vol. 6, 2018.

[12] E. Park, MS Lee, HS Kim and S. Bahk, "*AdaptaBLE*: Adaptive control of data rate, transmission power, and connection interval in bluetooth low energy", Computer Networks 181 (2020) 107520

[13] Michael Spörk, Carlo Alberto Boano, and Kay Römer . 2019. Performance and Trade-offs of the new PHY Modes of BLE 5. In ACMMobiHocWorkshop on Pervasive Systems in the IoTEra (PERSIST-IoT'19), July 2, 2019, Catania, Italy. ACM, New York, NY,

[14] I. M. Nikol'skii and K. K. Furmanov, "Efficiency of retransmission in sensor networks" Computational Mathematics and Modeling, Vol. 31, No. 4, October, 2020 DOI 10.1007/s10598-021-09508-z

[15] Seyed Mahdi Darroudi , Raül Caldera-Sànchez,Carles Gomez. "Bluetooth Mesh Energy Consumption: A Model". Sensors, 2019, 19, 1238

[16] Muhammad Rizwan Ghori, Tat-Chee Wan,Gian Chand Sodhy. "Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols". Sensors 2020, 20, 3590

[17] Jian Wang, Hui Liang. "Bluetooth 5.2 Technology and Application" Int. Journal of advanced Network, Monitoring and Controls, vol. 6, no. 2, 2021, pp. 32-36

[18] Junjie Yin, Zheng Yang, Hao Cao, Tongtong Liu, Zimu Zhou, Chenshu Wu. "A Survey on Bluetooth 5.0 and Mesh: New Milestones of IoT". ACM Trans. Sen. Netw. 2019, 15, 3, Article 28.

[19] A. Hernández-Solana, D. Pérez-Díaz-De-Cerio, M. García-Lozano, A. V. Bardají and J. L. Valenzuela, "Bluetooth Mesh Analysis, Issues, and Challenges," in IEEE Access, vol. 8, pp. 53784-53800. 2020.

[20] Martin Woodlley. "Understanding Reliability in Bluetooth Technology". Bluetooth. Version 1.0. 9 October 2020

[21] L. Almon, Flor Alvarez, L. Kamp, M. Hollick. "TheKing is Dead Long Life the King!. Towards Systematic Performance Evaluation of Heterogeneous Bluetooth Mesh Networks in Real World Environments". IEEE 44th Conf. on Local Computer Networks, 2019

[22] J. Silvestre-Blanes, J.C. García Ortiz, V.M. Sempere-Payá, D. Cuesta-Frau. "BLE Mesh using CODED PHY". II Interdisciplinary Conference on Mechanics, Computers and Electrics (ICMECE 2022), Barcelona, Spain.