Testing the interoperability of the joint scheduling and routing algorithm for IIoT applications

Andrej Hrovat[1], Gregor Cerar[1], Gordana Gardašević[2],
Dragan Vasiljević[2], Tomaž Javornik[1]

[1]Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] University of Banja Luka, Faculty of Electrical Engineering, Patre 5, 78000 Banja Luka, Bosnia and Herzegovina
e-pošta: andrej.hrovat@ijs.si

Abstract. *The interoperability challenge is of crucial importance in the IoT. In the paper a novel joint routing and scheduling algorithm which solves the problem of large number of queued packets and improves the network performance is presented and two IoT platforms for the benchmarking are described. The algorithm uses modified RPL protocol for transferring of routing information through the network and for network formation. The 6top protocol, defined in 6TiSCH standard is used for transmission slot scheduling between nodes. The novel algorithm will be implemented and tested by two 6TiSCH implementations, namely within OpenWSN deployed on Faculty of Electrical Engineering and Contiki-NG protocol stacks on LOG-a-TEC [6] testbeds deployed at JSI.*

1 Introduction

Industrial Internet of Things (IIoT) is a fast-growing technology that increases productivity and efficiency by combining the IoT platform with an existing industrial environment. While traditional wireless sensor network (WSN) applications have been assumed as delay-tolerant, real-time requirements and optimal scheduling are of utmost importance in industrial wireless networks. In addition, industrial applications assume large number of low-power wireless sensor devices spread in wide areas working in harsh multipath radio environment due to electromagnetic interference caused by surrounding machinery [1]. Therefore, the legacy wireless sensor communication technology standards must be adapted in order to mitigate time and frequency dependent interference and support delay sensitive applications. In this respect the Time-Slotted Channel Hopping (TSCH) standard has been developed for industrial automation and process control that uses time synchronization to achieve low-power operation and channel hopping to enable high reliability. Particularly, the 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4e) mechanisms are crucial for further adoption of IPv6 in industrial environments [2].

However, the standard does not specify or propose the particular scheduling and routing algorithm, whose design is left to researchers and vendors. In WSN, high end-to-end communication delay is caused by the non-adaptive scheduling to traffic demand. As a consequence, some nodes suffer from high number of queued packets which depends on routing topology and the number of scheduled transmission slots on particular node towards neighbouring nodes. In order to lower number of packets in the queue each node in 6TiSCH network can adaptively reserve additional transmission slots.

Since the mechanism for triggering the reservation of additional transmission slots (scheduling) is not defined by 6TiSCH standard, we propose a novel algorithm that joins routing and scheduling processes in order to solve the problem of large number of queued packets and improve the overall network performance in terms of packet delay, packet loss and throughput. The algorithm uses the modified RPL (Routing over Low Power and Lossy Networks) protocol adapted for transfer of routing information through the network, as well as for network formation. The 6top protocol, defined in 6TiSCH standard, is used for scheduling of transmission slots between nodes.

The algorithm is currently implemented within the OpenWSN [3] protocol stack by modifying the behaviour of RPL protocol which is considered the de facto routing protocol for the IoT. In the next step the proposed solution will be also validated within Contiki-NG [4] protocol stack.

Remaining of the paper is structured as follows. In the next section the IoT testbeds for implementation and testing of the proposed solutions are described. The third section gives the insight in the proposed joint scheduling and routing algorithm (JSRA). Finally concluding remarks with further work are given in last section.

2 IoT platforms

Since the importance of the interoperability challenge in the IoT, which arises due to the availability of diverse operating systems (OSs), programming languages, data structures, architectures and access mechanisms for things and data, has been emphasized by both academia and industry, the algorithm is benchmarked by two 6TiSCH implementations, namely within OpenWSN [3] and Contiki-NG [4] protocol stacks deployed on Faculty of Electrical Engineering (FEE) testbed and LOG-a-TEC [6] testbeds, respectively.

2.1 OpenMote-B IoT testbed

The main objective of this testbed is to establish a functional experimental platform for performing the measurement campaign in order to get the better insight into IIoT network operation, as well as to investigate the coexistence of 6TiSCH and RPL [7].

The testbed is comprised of 25 OpenMote-B devices and is set up at the Faculty of Electrical Engineering, University of Banja Luka [8]. The host computer is running Ubuntu with OpenWSN OS (latest release 1.22.0) and OpenVisualizer (OV) installed, see Fig.1 [9]. The OpenWSN is the first open implementation of protocol stack based on TSCH and other standardized protocols. As such, it represents a reference protocol stack for advanced research in IoT area. Moreover, the OpenWSN is the reference protocol stack for IETF 6TiSCH working group. OV is a software tool developed as a part of OpenWSN project that resides on host computer and provides communication and visualization functionalities. This enables the insight into status of connected motes, network management functions, provides sending commands to the node, etc. A Mosquitto broker that supports MQTT version 3.1/3.1.1 is installed, as well as additional Python applications with necessary libraries.

The OpenMote represents a new generation of open hardware platforms that is particularly adapted to the IIoT applications. Currently, a new board called OpenMote-B is being released, Fig.2. It provides a dual-radio interface for short and longer range communications, combined on one board. OpenMote-B incorporates the new Atmel AT86RF215 radio transceiver, which can operate simultaneously in the Sub-GHz (868/915 MHz) and the 2.4 GHz ISM bands, and supports all physical layer modes of the IEEE 802.15.4g standard (i.e., MR-FSK, MR-OFDM, MR-O-QPSK and OQPSK) with data rates ranging from 6 kbps to 2 Mbps. This is the first board that fully supports the IEEE 802.15.4g standard including MR-OFDM modulations for robust communications.

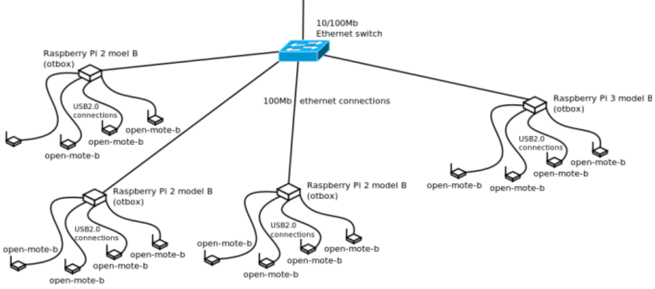

Fig. 1: One realization of 6TiSCH network within OpenMote-B IoT Testbed [9].

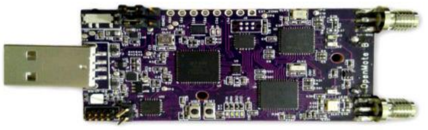

Fig. 2: OpenMote-B device [5].

2.2 LOG-a-TEC testbed

LOG-a-TEC [6] is comprised of several different radio technologies which enable dense and heterogeneous IoT, MTC and 5G experimentations. Specially developed embedded wireless sensor node can host four different wireless technologies and seven types of wireless transceivers.

In order to enable different experiments in combined outdoor/indoor environment using heterogeneous wireless technologies the testbed is deployed inside the JSI premises as well as outside in the park and on the walls of the buildings. The outdoor part of the wireless testbed is located at and around the park at JSI and is comprised of 56 nodes which are located over the area of 55 m by 60 m. Nodes are placed on light poles in the park 3.5 meters above the ground and on the surrounding walls at heights from 2 to 9.3 meters. To enable experiments in indoor environment and also outdoor/indoor influence analyzes the testbed is extended also in indoor environment with additional 20 UWB (Ultra-Wwide Band) nodes and one UNB (Ultra-Narrow Band) node (LoRa) which are deployed in second and third floor of the building with the dimensions of 28.4 m by 16.6 m.

The node locations with the available technologies per location are shown in Fig.3. On 26 possible locations 21 nodes with the TI CC1101 and AT86RF231 radios, 21 nodes with the TI CC2500 and AT86RF212 radios, 3 UNB LoRa nodes, 11 UWB nodes and 2 SNE – ESHTER (spectrum sensing) are deployed. The radio interfaces support experiments in frequency bands 860-1000MHz, 3.5 GHz to 6.5 GHz and 470-862 MHz [7].

LOG-a-TEC enables different experiments for studying the performance of the dense IoT and MTC (Machine-Type Communication) networks in the scope of the 5G networks. A wide range of radio interfaces supports experiments with heterogeneous MTC technologies by transmitting/receiving/sensing at the same time in the same location using beyond state of the art functionality like ultra-narrow band and ultra-wide band, packet based experimentation, clean slate protocol design, composable and modular protocol stacks, custom and advanced spectrum sensing and signal generating functions in sub-GHz spectrum.

In order to test newly developed JSRA algorithm IIoT applications the Contiki-NG operating system is ported on the nodes with the AT86RF231 and AT86RF212 radios. Contiki-NG is an operating system for resource-constrained devices which contains an RFC-compliant, low-power IPv6 communication stack, enabling Internet connectivity. Current implementation of TSCH and 6TiSCH supports so-called "minimal configuration", which only defines a basic RPL+TSCH network [10].

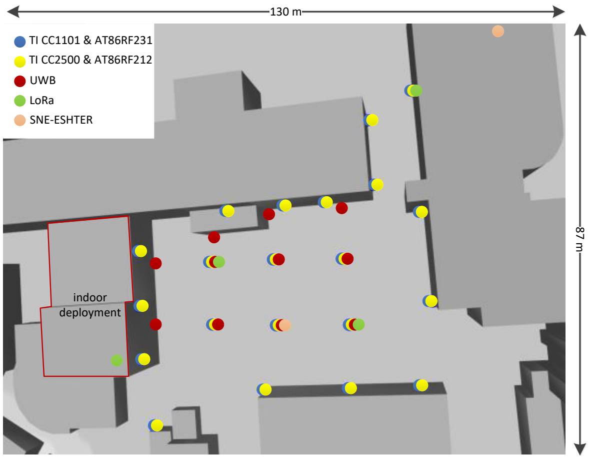

Fig 3. Locations of available radio interfaces in outdoor environment

3 Joint scheduling and routing algorithm

One approach to jointly consider the scheduling and routing operations is described in our paper [11].

In this paper, we have extended our research towards a novel framework for JSRA. The routing is imposed by topology at logical link-layer i.e. by scheduled transmission slots between neighbouring nodes, because routes can be established only over those nodes with scheduled transmission slots. On the other hand, finding the optimal route across the network requires the scheduling to be driven by routing protocol requirements. Since routing and scheduling are mutually dependent to achieve better network performance a cross-layer design approach can be used i.e. joining routing and scheduling mechanisms.

The packet delay, packet loss and throughput are metrics that can be used to assess behaviour of JSRA algorithm. The major conclusions from previous research are that in 6TiSCH network is not possible to achieve high network performances in terms of low packet loss, low packet delay and high throughput simultaneously [12], and that current RPL rank calculation based on ETX (Expected Transmission Count), as proposed by 6TiSCH minimal configuration [13], does not create stable and optimal routing topology [14].

Sensor nodes can host different applications where each application can have different communication requirements in terms of reliability, throughput and data transmission delay. Considering conclusions from previous research, the idea of the algorithm is to classify application data traffic into classes according to application communication requirements and jointly calculate routing and scheduling for each class separately using different routing metric than ETX. In that way we can achieve better network performance since routing and scheduling optimization per traffic class adapts the network to application requirements.

The algorithm uses hop count, RSSI (Received Signal Strength Indication) or LQI (Link Quality Indication) indicators and scheduled resources (what throughput, packet delay and packet reliability the parent node can provide) as a routing metrics. The node calculates its rank using those metrics. The rank is not calculated as a single value but as a rank vector where each value in the vector represent rank for certain traffic class. The parent nodes are chosen per traffic class as well. That means that the node in the network can have several parent nodes where each parent node corresponds to one or several traffic classes it can provide. So, multiple routes to network coordinator node are possible with scheduling along each route optimized for certain traffic class or possibly several classes.

The joint algorithm uses RPL protocol adapted for transfer of routing information through the network as a basis for routing as well as for network formation. RPL DIO (DODAG Information Object) packets will carry information to child nodes of hop count, parent rank vector and scheduled resources the parent node can additionally provide. Thus, additional fields in the DIO packet are defined and used. RPL protocol [15] in its data structures provides some fields which are not currently in use. Those fields are used for this purpose.

For scheduling of transmission slots between nodes the 6top protocol [16], defined in 6TiSCH standard, is used. A node in the network can calculate its traffic requirements which is sum of traffic requirements of applications residing on that node and traffic requirements of its child nodes. Based on its traffic requirements and information about available scheduled resources received via DIO from parent nodes, the node can initiate the timeslot scheduling towards one or more parent nodes (depending on what each parent node can provide of scheduled resources) using 6top protocol. The scheduling continues upwards along the routes for traffic classes of interest all the way to the network coordinator. This represents a resource reservation process. This process ensures that available network resources are optimized for application traffic requirements which increases overall network performance. To support this process each application has to declare what traffic class it generates and what are its throughput requirements.

The network establishment process starts when network coordinator node i.e. DAG (Directed Acyclic Graph) root starts broadcasting EB packets at MAC layer and DIO packets at routing layer. In DIO packets it offers entire slot frame as available resources for scheduling. The nodes in range of DAG root first synchronize to network by listening for EB according to 6TiSCH minimal configuration [12]. Once synchronized on MAC layer the nodes are then listening for DIO packets. After receiving DIO, the nodes initiate timeslot assignment i.e. scheduling with DAG root according to traffic requirements of applications residing on the nodes. When scheduling is finished DAG root will update information in DIO about available scheduled resources it can provide. Based on information from updated DIO and possibly resources that are not fully used by the applications residing on the newly joined nodes, they will calculate their rank vector and start broadcasting their own DIO with information about scheduled resources they can provide. In the same time, they will create routes to DAG root for the required traffic classes. The process continues until all nodes join the network. When some node joins the network and schedule resources to its parent nodes, the parent nodes will have to initiate resource scheduling upward along routes for traffic classes required by joined child node. The scheduling continues all the way to DAG root node whereby each node shell first tries to use what it already has scheduled and not fully used and then schedule new resources.

If the parent node can not provide required resources to the child node, the child node will be able to join the network with only best effort traffic class available. The best effort traffic class is always available in the network.

The exchange of DIO and DAO (Destination Advertisement Object) will be intensive during network establishment phase since each node that joins the network triggers resource scheduling and resource scheduling triggers node rank vector recalculation and updating of relevant information in DIO. Once the network is established the amount of DIO and DAO will be significantly smaller.

4 Conclusions

In the paper the framework of the JSRA algorithm is described and the details of the two IoT platforms with the OS supporting 6TiSCH are given. In particular, the coexistence of the 6TiSCH and RPL will be investigated by OpenMote-B IoT testbed with the OpenWSN and LOG-a-TEC testbed with the Contigi-NG.

Since routing and scheduling are mutually dependent, across-layer design approach is proposed to achieve better network performance, i.e. joining routing and scheduling mechanisms. In order to achieve low packet loss, low packet delay and high throughput simultaneously the algorithm classify application data traffic into classes according to application communication requirements and jointly calculate routing and scheduling for each class separately.

Further work will incorporate the implementation of the JSRA algorithm on OpenWSN and Contigi-NG protocol stack for the benchmark and interoperability testing.

Acknowledgment

This work was funded by the Slovenian Research Agency under the grants no. BI-BA/19-20-045 and no. P2-0016.

References

[1] C. G. Vehbi, G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches." IEEE Transactions on industrial electronics 56.10 (2009): 4258-4265.

[2] IETF, Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. https://tools.ietf.org/html/rfc7554.

[3] OpenWSN project. http://www.openwsn.org/.

[4] Contiki-NG. https://github.com/contiki-ng.

[5] OpenMote, [online] Available: http://www.openmote.com/.

[6] http://log-a-tec.eu/.

[7] http://sensorlab.ijs.si/hardware.html.

[8] http://www.communications.etfbl.net/en_labTEL.html

[9] G. Gardašević, P. Plavšić, D. Vasiljević, "Experimental IoT Testbed for Testing the 6TiSCH and RPL Coexistence", IEEE INFOCOM 2019, CNERT Workshop, Paris, France, April 2019.

[10] S. Duquennoy, A. Elsts, B. A. Nahas and G. Oikonomo, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, 2017, pp. 11-18.

[11] G. Gardašević, D. Vasiljević, C. Buratti, and R. Verdone, “Experimental Characterization of Joint Scheduling and Routing Algorithm Over 6TiSCH,” in 2018 European Conference on Networks and Communications (EuCNC), June 2018, pp. 424-428.

[12] D. Vasiljević and G. Gardašević, "Performance evaluation of OpenWSN operating system on open mote platform for industrial IoT applications," 2016 International Symposium on Industrial Electronics (INDEL), Banja Luka, 2016, pp. 1-6. doi: 10.1109/INDEL.2016.7797791.

[13] X. Vilajosana, K. Pister, and T. Watteyne, “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration,” BCP 210, RFC 8180, DOI 10.17487/RFC8180, May 2017, https://www.rfc-editor.org/info/rfc8180.

[14] R. T. Hermeto, A. Gallais, F. Theoleyre, “On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment,” Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 83 – 90, Montreal, QC, Canada — October 28 - November 02, 2018.

[15] T. Winter, P. Thubert, A.Brandt, J.Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[16] Q. Wang, X. Vilajosana, and T. Watteyne, “6TiSCH Operation Sublayer (6top) Protocol (6P),” RFC 8480, DOI 10.17487/RFC8480, November 2018, <https://www.rfc-editor.org/info/rfc8480>.