Network Formation in 6TiSCH Industrial Internet of Things under Misbehaved Nodes

Yassine Boufenneche, Rafik Zitouni, Laurent George, Nawel Gharbi

▶ **To cite this version:**

Yassine Boufenneche, Rafik Zitouni, Laurent George, Nawel Gharbi. Network Formation in 6TiSCH Industrial Internet of Things under Misbehaved Nodes. 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Dec 2020, Paris, France. pp.1-6, 10.1109/IOTSMS52051.2020.9340200 . hal-03570206

HAL Id: hal-03570206
https://hal.science/hal-03570206v1

Submitted on 13 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Network Formation in 6TiSCH Industrial Internet of Things under Misbehaved Nodes

Yassine Boufenneche*, Rafik Zitouni‡, Laurent George** and Nawel Gharbi*
MOVEP Laboratory*, Department of Computer Science, USTHB, Algiers, Algeria
ECE Paris Research Lab‡, 37 Quai de Grenelle, 75015 Paris, France
LIGM**, Université Gustave Eiffel/ESIEE Paris, Paris, France
Email: yboufenneche@usthb.dz, rafik.zitouni@ece.fr, laurent.george@esiee.fr, ngharbi@usthb.dz

***Abstract*—Time Slotted Channel Hopping (TSCH) is a new Medium Access Control (MAC) protocol proposed by the IEEE 802.15.4e standard. It is designed to meet the requirements of industrial networks, such as high Packet Delivery Ratio (PDR) and bounded delays, along with low energy consumption. TSCH is now the basis of a full stack for Industrial Internet of Things (IIoT) proposed by the International Engineering Task Force (IETF), known as 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4e). Since 6TiSCH networks are expected to offer high performance and fast bootstrapping, the network formation time could be impacted by the network size and the rate of control packets. In this paper, we demonstrate that non cooperative nodes, which can be malicious, could also drastically increase the network joining time. First, we propose the attack model and its implementation on the 6TiSCH simulator. Then, we carry out a set of experiments for different network sizes. Finally, we show through simulation results the impact of the proposed attack on the joining time.**

***Index Terms*—Industrial Internet of Things (IIoT), 6TiSCH, 6top, Performance Evaluation, Joining Time, Cyber Security, IEEE 802.15.4e, MAC protocols**

I. Introduction

WirelessHART [1] and ISA100.11a [2] are two pioneer wireless technologies able to provide reliable communication for industrial automation [3]. However, these are proprietary solutions, and they are also not IP compliant, which prompted standardization bodies like IEEE and IETF to develop new open and standard solutions. In a first time, the IEEE proposed in its 802.15.4e standard [4] TSCH, a MAC protocol based on Time Division Multiple Access (TDMA) and channel hopping, to allow deterministic channel access and also to mitigate the effect of signal interference. Later, the IETF adopted the TSCH protocol and the IEEE 802.15.e PHY as lower layers, and developed a set upper layers, resulting in a full stack enabling IP communications to small sensors and actuators in industrial environments.

In TSCH, time is organized in slotframes, each one is made up of a set of timeslots. A timeslot must be large enough so that two neighboring nodes can exchange a data packet and its acknowledgment. Communications of each node are orchestrated by a TSCH schedule, which tells the node exactly when to send its own frames and when to receive frames from its neighbors. The schedule is represented by a matrix whose cells are identified by a timeslot number and a channel offset. We point out that there are two kinds of cells (slots): shared ones, which are used to exchange control messages, and dedicated ones, used for sending and receiving data between neighboring nodes. Figure 1 depicts an example of a TSCH schedule. The cell (1,1) represents a shared slot. Node B uses slots (2,1) and (7,4) to send data to node A, and it expects to receive data from nodes D and E on slots (3,3) and (6,1), respectively.

For network formation, nodes follow a joining process and exchange periodically control messages like Enhanced Beacon (EB) frames, Join Request (JR) frames and DIO (DODAG Information Object) messages on shared slots. Enhanced Beacons are sent by already joined nodes, join requests frames are sent by the new joining nodes, while DIOs messages are used for the routing topology construction. The delay required for a node to synchronize with the network, identify a routing parent, reserve a dedicated slot to communicate with its parent and start sending DIO packets is called the joining time [5].

The joining procedure proposed in 6TiSCH networks engender higher joining times, which depend on the network size and the time resources allocated for control messages [6], [7]. However, network nodes behavior also could protract the joining time. In fact, nodes may opt for an abnormal or non cooperative behavior willingly or unintentionally. In the former case, nodes decide not to cooperate with their neighbors when carrying out network operations, to save their resources like energy and bandwidth. In the latter case, nodes operation is altered by external malicious entities intending to make a stealth attack through legitimate but corrupted nodes. In the rest of this paper, we refer to both kinds of nodes by misbehaving or cheating. The both adjectives or words will be used interchangeably throughout the paper.

In our work, we consider the problem of non cooperative nodes in 6TiSCH networks and their impact on the network formation. We summarize our major contributions as follows:

- We define a 6TiSCH-specific non cooperative behavior attack. The attack is further implemented in the 6TiSCH simulator [8].
- We carry out a set of simulations in a 6TiSCH network in the presence of misbehaving (non cooperative) nodes.
- We study the impact of a such an attack on the joining time.

The rest of the paper is organized as follows. Section II highlights the 6TiSCH main protocols and vulnerabilities.

Then, we analyze the joining process in 6TiSCH networks, and we give some relevant related works in section III. Section IV describes the behavior of non cooperative nodes, while section V presents the simulation setup and discusses our findings. Finally, section VIconcludes the paper and gives some directions for future work.

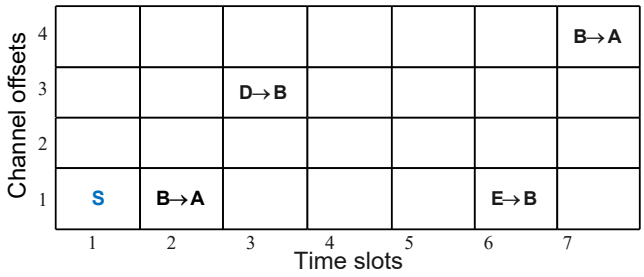

Fig. 1. Example of a TSCH schedule.

II. 6TiSCH STACK VULNERABILITIES

Cyber security attacks in IoT have been classified into two categories: *passive attacks* and *active attacks* [9]. The first class is related to eavesdropping, node malfunctioning, node outage and traffic analysis. In this case, the threat is hidden and the nodes' intention is to collect useful information. By contrast, in the second class, the attacks are operated through jamming and spoofing (Sybil attack) causing a Denial of Service (DoS). Non cooperative nodes' behavior is considered as an active attack since the role of some nodes is altered by external entities. Bellow, we highlight four analyzed active attacks against 6TiSCH IoT networks.

a) Attacks towards TSCH: As mentioned in section I, time is split in timeslots. Each one is identified by a number called Absolute Slotframe Number (ASN). The frequency used when two nodes communicate during a timeslot is a function of the ASN and a channel offset. Attackers may use this information to carry out ASN attacks, which consist of sending fake EBs with incorrect ASNs. Using a wrong ASN prompts the victim to compute an incorrect frequency, which would make it unable to communicate with its neighbors [10]. Another possible attack on TSCH is the TSCH de-synchronization, where an attacker transmits the messages on the slots that are allocated to the other users. This engenders packet collisions and losses, and carrying out series of these attacks could cause the de-synchronization of neighboring nodes [9].

b) Attacks towards 6LoWPAN: 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Network) is an adaptation layer that enables resource constrained objects to interact with IPv6 devices. It offers header compression and fragmentation mechanisms [11]. Taking advantage of the absence of a verification process, attackers can carry out fragmentation attacks by putting their own fragments in the fragmentation chain of an IPv6 packet, and without being detected as spoofed or duplicated fragments [9], [12].

c) Attacks towards RPL: RPL (Routing Protocol for Low power and Lossy Networks) [13] is the routing protocol adopted in the 6TISCH stack. It relies on a ranking mechanism to build a routing typology called Destination Oriented Directed Acyclic Graph (DODAG) (see Figure 2). The root starts broadcasting DIO messages to its neighbors, which will in turn propagate their own DIOs later. Nodes receiving the DIOs select the neighbor with the smallest rank as a parent. A node can request the transmission of DIO messages from its neighborhood by sending DIS (DODAG Information Solicitation) messages. RPL may undergo a multitude of attacks, including rank attack and DIS attack. In the rank attack, a malicious node changes its rank to incite neighbors to select it as a parent. This may result in routing loops, un-optimized paths and more control overheads. DIS attack consists in periodically sending DIS messages by the attacker to delude legitimate nodes that there are some nodes willing to join the network, or something went wrong with the topology. The aim of this attack is to increase the control overhead and hence the energy exhausting [12].

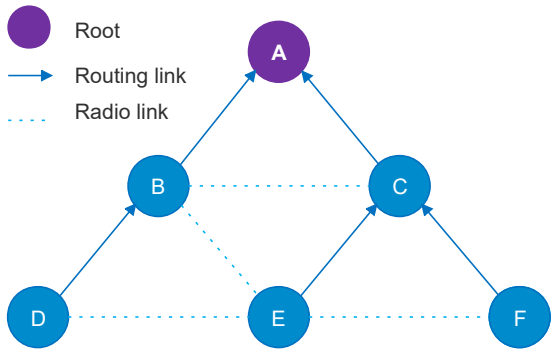

Fig. 2. Simple routing topology.

d) Attacks towards 6top: The 6TiSCH Operation Sublayer (6top) is made up of a scheduling function like MSF (Minimal Scheduling Function) [5] and a 6top Protocol (6P) [14]. The role of the scheduling function is to adapt the number of cells scheduled by the nodes to the traffic. Regarding 6P, it offers a negotiation process called 6P transaction, which enables two neighbors to agree upon a certain schedule (*e.g.* which cells to add to both schedules of the neighbors). The negotiation process consists in exchanging dedicated messages (requests and responses) between two neighbors, and ensures that they have common cells in their schedules. Carignani et al. [15] proposed a threat model of 6P, where an adversary intercepts and forges 6P messages to carry out bogus 6P transactions between two neighboring nodes. This eventually results in incoherent scheduled cells between the two peers, and hence no communication between them is possible. In section IV we propose another threat model related to 6P.

III. Analysis of the Joining Process

Nodes join a 6TiSCH network following a formation process starting by enabling a single-hop communication between neighbors, and ending by the construction of the RPL toplogy (see Fig. 2) [6].

A node that intends to join the network starts by listening for Enhanced Beacons during a predefined period of time or until it receives EBs from a known number of neighbors. Then, it selects a neighbor as a Join Proxy (JP) to continue the joining process. After having selected a JP, the node sends a Join Request to the Join Registrar/Coordinator (JRC) through its JP, and becomes a joined node when it receives a Join Response from the JRC. The joined node receives DIOs, computes its own rank, and selects a routing parent, with which it must install a negotiated cell through a 6P transaction. At this stage, the joined node can start sending EBs and DIOs in turn.

There are few works in literature which studied the time needed for the joining process, known as the joining time [3], [6], [7], [16], [17]. Authors in [16] proposed a random-based advertisement algorithm, where nodes send EBs with a certain probability, an they studied the impact of the number of channels used for advertising on the joining time. They found that increasing the number of channel offsets reduces significantly the joining time. The work in [6] investigated the network formation in 6TiSCH networks. It was shown that the standard strategy for the allocation of timeslots to control messages, in which a single static shared timeslot is used, engender long network formation delays in large scale networks. Authors proposed the use of dynamic allocation strategy for shared timeslots that depends on number of control messages to guarantee a fast joining process. Authors in [17] also investigated the network formation process and they proposed a dynamic beacon interval which depends on the number of joined nodes. A dynamic allocation strategy has been also proposed in [3], that aims to reduce the joining time. Nodes locally adapt the number of slots for transmitting control messages. Authors in [7] showed that permanently assigning EB frames higher priority over other control messages like DIOs could increase the joining time of the new joining nodes. Therefore, they propose to give higher priority to DIOs when new nodes need DIO packets to complete their joining process.

All the above-mentioned works focus on the limitations of the strategy for the allocation of timeslots to control messages, especially in large deployments. Furthermore, they consider that the network is safe, and there are no malicious nodes. In our work, we are interested in the joining time when network contains corrupted nodes.

IV. Misbehavior Model

In this section, we introduce the conduct of misbehaving nodes towards their neighbors. As mentioned in section I, 6TiSCH nodes exchange data packets during dedicated slots. If one node has more data packets to send, but the actual resources (slots) are not satisfactory, it should ask its neighbor for additional resources. For so doing, the two parts agree on new slots and add them to their schedules. The first node initiates a 6P transaction by sending a request containing the number and the proposed positions of slots. The neighbor sends back a response containing the slots that it agrees for addition. If the neighbor is a cooperative node, its response depends only on the availability of the proposed slots in its schedule. However, if it is a cheating one, it refuses to answer the request, either to save its limited resources or to engender higher delays or packet losses if it is a question of a corrupted node.

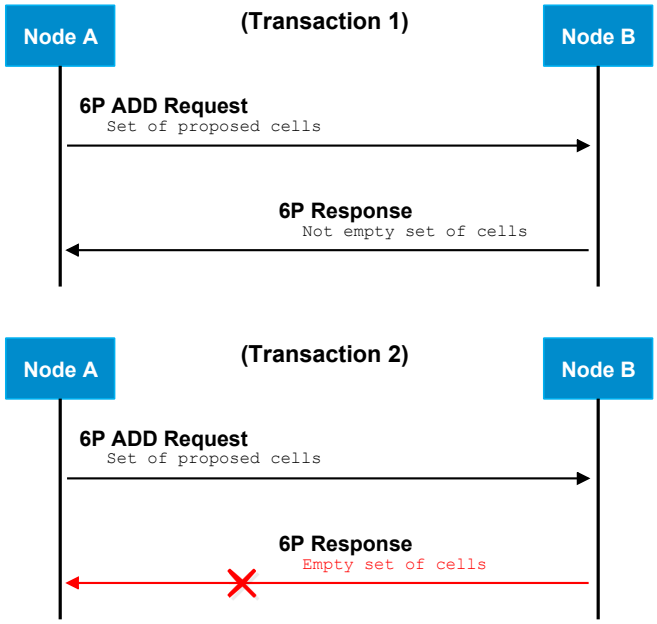

Fig. 3. Successful and unsuccessful transactions.

Figure 3 demonstrates two 6P transactions with honest and misbehaving nodes. The first one is executed between two authentic nodes. Node A issues the transaction by sending an ADD request to node B, which responds back with the available and validated set of cells. Then, the node A updates its schedule by adding these cells for upcoming communications with the node B. In the second transaction, the node B is lying, by answering with an empty list of cells, *i.e.* it didn't agree the request of the node A.

A cheating node may decide not to agree all new requests from all neighbors. In this case, it will be easily spotted by prospective preventing processes. For some caution, the cheating node may accept to agree the first request for each neighbor.

The conduct of a misbehaving node is described in Algorithm 1. The misbehaving node maintains a set, initially empty, containing the IDs (identifiers) of neighbors that have already completed successful ADD transaction with this misbehaving node. When the misbehaving node receives a new transaction, it determines its type. If it is an ADD request, it first makes sure that the set $ServedNeighbors$ does not contain the neighbor's ID before it agrees the request and adds the ID

to $ServedNeighbors$. Otherwise, the misbehaving node does not agree the request.

Algorithm 1: Conduct of a misbehaving node

```
Input: ServedNeighbors = ∅ // set of
       neighbors already received a
       response
wait for a new transaction
RequestType = get the request type of the current
  transaction
if RequestType = ADD then
    ID = get the neighbor ID
    if ID ∉ ServedNeighbors then
        agree the neighbor's request
        add ID to ServedNeighbors
    else
        deny the neighbor's request
```

The misbehavior of the root is expected to have greatest impact, and therefore, in the simulation scenario, we set the root as a misbehaving node only if all nodes are misbehaving (i.e. the misbehavior rate is 100%). In our strategy of selecting misbehaving nodes, we start with the first nodes that have successfully joined the network, and therefore, the closest nodes to the root. Algorithm 2 describes this selection strategy. At the beginning, the rate of already configured cheating nodes ($CurrentRate$) is equal to 0. If the desired misbehaving rate is 100%, the root is configured to be misbehaving. Otherwise, the algorithm continues to set the newly joining nodes as misbehaving ones until the value of $CurrentRate$ reaches the value of $MisbehaviorRate$.

Algorithm 2: Choosing misbehaving nodes

```
Input: N // Number of all nodes
Input: MisbehaviorRate
CurrentRate ← 0
// Rate of already configured
  misbehaving nodes Counter ← 0
if MisbehaviorRate = 100 then
    Set the root as a misbehaving node
else
    while CurrentRate < MisbehaviorRate do
        Wait till a new node X joins the network
        Set X as a misbehaving node
        Counter ← Counter + 1
        CurrentRate ← Counter/N
```

V. Simulation

To study the impact of such an attack on the joining time, we first implemented the node's misbehavior at the 6top sublayer, then we integrated it in the 6TiSCH simulator, a dedicated tool for 6TiSCH networks. We considered different network sizes: 20, 60 and 100 nodes. The nodes generate periodic traffic, and send their packets upwards to the root in a multi-hop fashion. A single simulation is carried out during 15000 slotframe cycles (252.5 minutes), and each slotframe comprises 101 timeslots. Every simulation is repeated three times. Table I summarizes the main parameters used to perform simulations. We changed the misbehaving rate in the network, and we repeated the simulation for each rate. A rate equal to 0% means that no node is misbehaving in the network. Likewise, a rate equal to 100% indicates that all nodes are misbehaving.

TABLE I
Main simulation parameters

Parameter	Value
Number of nodes	20, 60, 100
Simulation time	15000 slotframes
Slotframe size	101 slots
Application type	Periodic
Packet generation period	4 (s)
Packet length	90 (octets)
RPL Objective Function	Objective Function 0 (OF0)
TSCH slot duration	0.010 (s)
Scheduling function	Minimal Scheduling Function
Physical channels	16

The subsequent subsection report the obtained results in terms of the network joining time. Figure 4 presents the average joining time as a function of the number of misbehaving nodes, considering three different network sizes. We can see clearly through the curves that the joining time in small size networks (20 nodes) is not impacted by the number of misbehaving nodes; it remains around 500 seconds whatever the misbehaving rate. We can explain this by the fact that small networks generate low traffic, and nodes don't need to request more dedicated timeslots to forward the data. For large deployments (60 nodes and 100 nodes), the joining time is an increasing function of the number of misbehaving nodes. For instance, when 60% of nodes are misbehaving, we notice an increase of 27% and 63% respectively when the network comprises 60 nodes and 100 nodes. When all nodes are misbehaving, including the root, the increase of the joining time reaches 128% and 174% respectively in networks with 60 nodes and 100 nodes. The curves also show that the network size affects the joining time; larger networks manifests higher joining times.

Figure 5 emphasizes the distribution of the individual joining times for every node and confirm the obtained results in Figure 4. One box represents the joining times of all nodes in the network given a misbehavior rate. The yellow line inside a box corresponds to the median, while the upper and bottom lines are the maximum and the minimum values, respectively. The points or small circles outside the boxes are the outliers values. For the different plots (A), (B) and (C), the worst joining time is correlated to the number of cheating nodes. By increasing the number of nodes, the obtained joining time is longer when the misbehavior rate is 80 %. The worst joining

time is found equal to 6000 seconds (outlier point), and it would be much longer if we have increased the network size. When the nodes cheat giving a wrong number of available cells, other nodes spend additional time waiting for a good schedule.

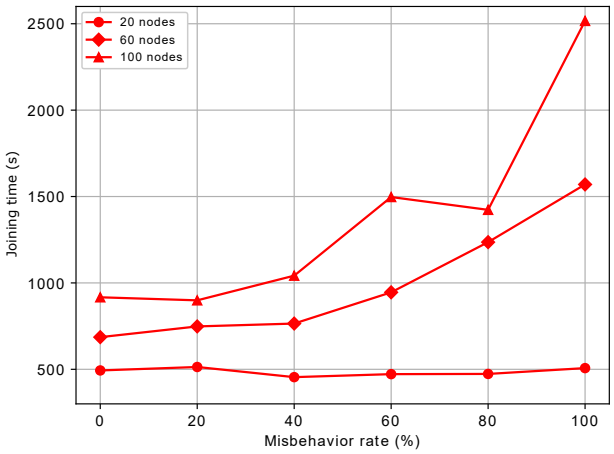

Fig. 4. Obtained joining time of SA-6TiSH compared to 6TiSH under different misbehavior rates.

VI. Conclusion

We investigated in this paper the joining time in 6TiSCH networks, since fast network formation is expected in industrial environments. First, we introduced the concept of non cooperative nodes in the 6P protocol, which can be considered as a kind of attack. Non cooperative nodes might deny the requests received from neighbors aiming to negotiate more resources, simply because they want to save their energy, or because they intend to disrupt the network performance in the case of compromised nodes. We implemented the proposed model on the 6TiSCH simulator, then we performed a series of experiments. The obtained results demonstrated that the proposed attack is not effective in small networks unlike in large scale networks, where the joining time is proportional to the number of misbehaving nodes. In fact, the increase in the joining time could reach 174% when considering high misbehaving rates. Furthermore, simulation results showed that augmenting the network size also increases the joining time.

In our future work, we plan to propose a countermeasure against cheating and lying nodes. An ongoing work has been initiated using the fuzzy logic decision maker to measure the credibility of nodes' messages.

References

[1] *Industrial networks - Wireless communication network and communication profiles - WirelessHART*, 2016.

[2] *Wireless systems for industrial automation: Process control and related applications*, 2011.

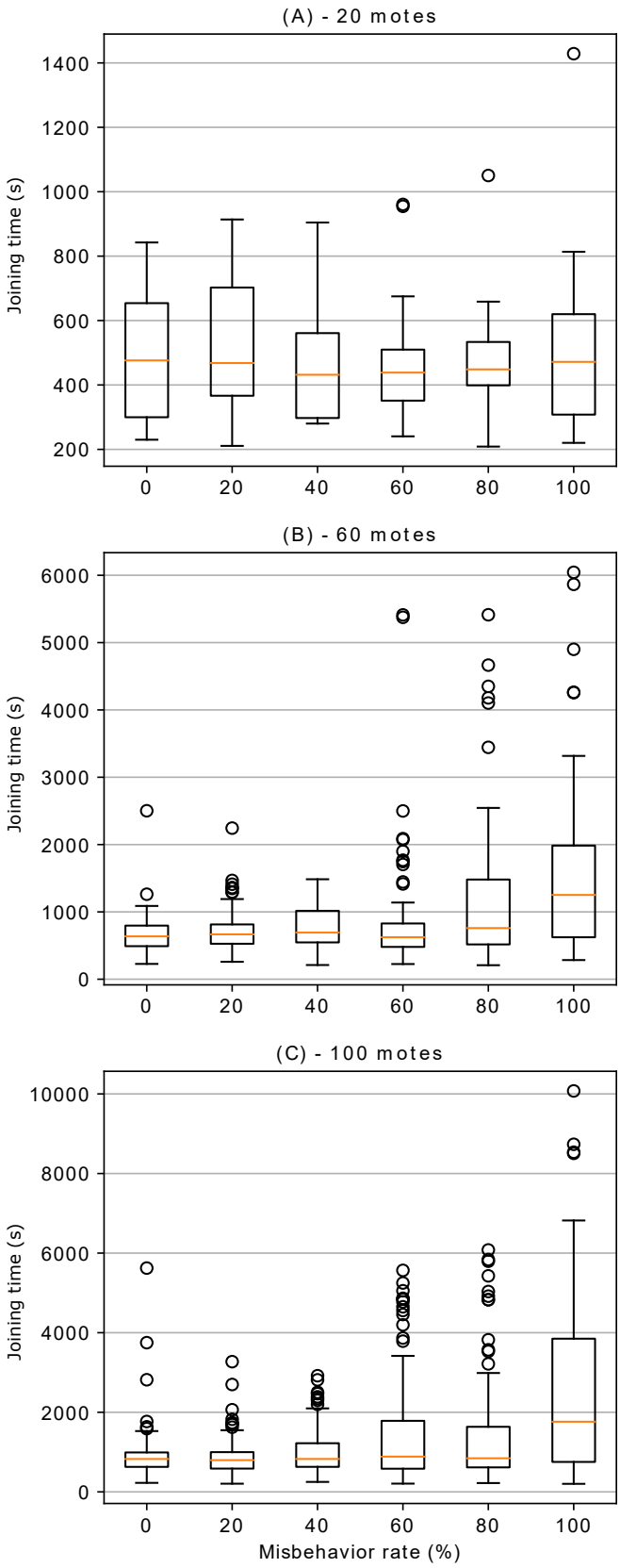

Fig. 5. Impact of misbehavior on the joining time for each node in the network.

[3] D. Fanucchi, B. Staehle, and R. Knorr, "On the suitability of 6tisch for industrial wireless communication," in *Kommunikation und Bildverarbeitung in der Automation*. Springer Vieweg, Berlin, Heidelberg, 2020, pp. 34–48.
[4] *IEEE Standard for Low-Rate Wireless Networks*, April 2016, IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011).
[5] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-msf-18, 2020, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-6tisch-msf-18
[6] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving network formation in 6tisch networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 98–110, 2018.
[7] A. Kalita and M. Khatua, "Opportunistic priority alternation scheme for faster formation of 6tisch network," in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, 2020, pp. 1–5.
[8] E. Municio, G. Daneels, M. Vučinić, S. Latré, J. Famaey, Y. Tanaka, K. Brun, K. Muraoka, X. Vilajosana, and T. Watteyne, "Simulating 6tisch networks," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, p. e3494, 2019.
[9] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.
[10] M. SMACHE, A. OLIVEREAU, T. FRANCO-RONDISSON, and A. TRIA, "Autonomous detection of synchronization attacks in the industrial internet of things," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, 2019, pp. 1–9.
[11] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, Sep. 2007. [Online]. Available: https://tools.ietf.org/html/rfc4944
[12] P. Pongle and G. Chavan, "A survey: Attacks on rpl and 6lowpan in iot," in *2015 International Conference on Pervasive Computing (ICPC)*, 2015, pp. 1–6.
[13] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, R. K. Alexander *et al.*, "Rpl: Ipv6 routing protocol for low-power and lossy networks." *rfc*, vol. 6550, pp. 1–157, 2012.
[14] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," RFC 8480, Nov. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8480.txt
[15] G. Carignani, F. Righetti, C. Vallati, M. Tiloca, and G. Anastasi, "Evaluation of feasibility and impact of attacks against the 6top protocol in 6tisch networks," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2020, pp. 68–77.
[16] D. De Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, "A performance analysis of the network formation process in ieee 802.15.4e tsch wireless sensor/actuator networks," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, 2014, pp. 1–6.
[17] A. Kalita and M. Khatua, "Faster joining in 6tisch network using dynamic beacon interval," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2019, pp. 454–457.

View publication stats