Effect of DIS Attack on 6TiSCH Network Formation

Alakesh Kalita, *Student Member, IEEE*, Alessandro Brighente, Manas Khatua, *Member, IEEE*, and Mauro Conti, *Fellow, IEEE*

***Abstract*— The 6TiSCH standard provides minimum latency and reliability in mission-critical IoT applications. To optimize resource allocation during 6TiSCH network formation, IETF released the 6TiSCH minimal configuration (6TiSCH-MC) standard. 6TiSCH-MC considered IETF's IPv6 Routing Protocol for Low power and Lossy network (RPL) as a routing protocol for both upward and downward routing. In RPL, new joining nodes or joined nodes transmit DODAG Information Solicitation (DIS) requests to get routing information from the network. However, we observe that malicious node(s) can severely affect 6TiSCH networks by sending multiple DIS requests. In this letter, we show and experimentally evaluate on real devices the impact of the DIS attack during 6TiSCH networks formation. We show that the attacker does not need expensive resources or access to the network's sensitive information to execute the *DIS attack*. Our testbed experiments show that the DIS attack significantly degrades the nodes' joining time and energy consumption, increasing them by 34% and 16%, respectively, compared to normal functioning during 6TiSCH network formation.**

***Index Terms*— 6TiSCH, DIS attack, network formation, RPL attack.**

I. Introduction

THE IPv6 over IEEE 802.15.4e Time Slotted Channel Hopping (6TiSCH) standard is designed to provide reliable and delay-bounded multi-hop communication in low power and lossy networks (LLNs) using the existing IPv6 network infrastructure [1], [2]. IETF's 6TiSCH working group released the *6TiSCH minimal configuration* (6TiSCH-MC) standard [3] for 6TiSCH network bootstrapping. The routing protocol used by 6TiSCH-MC, i.e., Routing Protocol for Low power and Lossy network (RPL) [4], is exposed to various security threats, like selective forwarding attack, sinkhole attack, sybil attack, hello flooding attack, and wormhole attacks [5], [6]. Among these attacks, *DODAG Information Solicitation* (DIS) *attack* targets the service availability, specifically, the energy consumption of the legitimate nodes and the exhaustion of the network bandwidth. Although DIS attack has been studied in the literature [7]–[9], its effect was not studied on the 6TiSCH network, specifically on 6TiSCH network formation. As the existing works on DIS attack are mainly based on the previous version of IEEE 802.15.4 [10], they did not consider that the control packets are transmitted in shared cells only. In fact, in 6TiSCH-MC only one cell (known as *shared cell* or *minimal cell*) in a *slotframe*[1] should be used for exchanging all types of bootstrapping control traffic, such as Enhanced Beacon (EB), RPL control packets [4], Join Request (JRQ), and Join Response (JRS) frames. The existing works considered enough bandwidth for the transmission of all the generated DIO packets due to DIS attack. However, we observe that DIS attack can severely congest the shared cell in 6TiSCH networks by forcing the nodes to contend for shared cell at the same time.

In this letter, we thoroughly study the DIS attack on 6TiSCH network using testbed experiments. Our testbed comprises a *root node*, a *malicious node*, and 10 other *6TiSCH nodes* which join the network one by one. We investigate the effects of the DIS attack in terms of *joining time*, *energy consumption* of the nodes, and *stability* of the network. Our results show that, without consuming much resources, a malicious node can significantly increase the network formation time and energy consumption of the other nodes while creating an inconsistent/unstable network. During our testbed experiments, the DIS attack degrades the joining time, increasing it by 34% compared to normal behaviour. Furthermore, it increases the nodes energy consumption by 16%. In brief, the major contributions of this work are as follows.

- We show the feasibility of DIS attack and its effect on 6TiSCH network formation. To the best of our knowledge, this is the first letter showing the feasibility and impact of this attack in 6TiSCH network.
- We perform testbed experiments to show how DIS attack can degrade the joining time, energy consumption of the nodes, and stability of 6TiSCH network.

The rest of the letter is organized as follows. Section II describes the procedure for 6TiSCH network formation and DIO generating Trickle algorithm. Section III briefly describes the DIS attack on RPL based 6TiSCH network. Section IV describes the testbed experimental evaluation of DIS attack on 6TiSCH network formation. Finally, Section V concludes the letter.

Manuscript received November 26, 2021; revised January 30, 2022; accepted February 26, 2022. Date of publication March 2, 2022; date of current version May 10, 2022. The authors received partial financial support from SERB, India, for building the research testbed used in this article. The associate editor coordinating the review of this letter and approving it for publication was D. Cassioli. *(Corresponding author: Alakesh Kalita.)*

Alakesh Kalita and Manas Khatua are with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India (e-mail: alakesh.kalita1025@gmail.com; manaskhatua@iitg.ac.in).

Alessandro Brighente is with the Department of Mathematics and HIT Research Center, University of Padova, 35122 Padova, Italy (e-mail: alessandro.brighente@unipd.it).

Mauro Conti is with the Department of Mathematics and HIT Research Center, University of Padova, 35122 Padova, Italy, and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft, 2628 CD Delft, The Netherlands (e-mail: conti@math.unipd.it).

Digital Object Identifier 10.1109/LCOMM.2022.3155992

[1] Slotframe is a collection of *timeslots* that repeats over time. A timeslot is long enough (typically 10 ms) to transmit a packet and receive its acknowledgement.

II. Background

In this section, we provide an overview of the main concepts needed to understand the DIS attack and its effects on 6TiSCH networks. We first describe the process that leads to 6TiSCH network formation in Section II-A, and then describe the algorithm that governs the DIO packet generation and transmission in Section II-B.

A. Formation of 6TiSCH Networks

The join registrar/coordinator (JRC) or RPL root node initiates the formation process by periodically broadcasting EB frames. An EB frame contains the basic information of a network, such as the JRC-id, the duration of a timeslot, the number of timeslots in a slotframe, and the channel hopping sequence. When pledges want to join the network, they are unaware of the channel and time when already joined nodes transmit their control packets. Pledges hence periodically scan each channel for a fixed amount of time. When a pledge receives a valid EB from an already joined node, it becomes a *TSCH synchronized node*. If the pledge receives several EBs from different joined nodes, it selects its parent based on the value in the `Join Metric` field of the received EB frames. The pledge synchronizes within the shared timeslot, and remains idle in the other slots to save energy. To complete the joining process, the TSCH synchronized node needs to receive a valid DIO packet, which is *multicasted/unicasted* by the RPL routing layer of the joined nodes. The DIO packet contains all the necessary routing information required to join the DODAG routing tree constructed by the RPL. While waiting for the DIO packet, the TSCH synchronized node exchanges JRQ and JRS control frames with the JRC for *secure enrollment*. Once the secure enrollment is done and the TSCH synchronized node receives a valid DIO packet, the TSCH synchronized node becomes a *6TiSCH joined node* or *RPL joined node*. The formation of the entire network gets completed when all the pledges complete their joining process.

Note that, according to the 6TiSCH-MC standard [3], all types of control packets need to be transmitted/exchanged by all nodes in the shared cell of a slotframe. Hence, nodes need to contend for the channel access associated with the shared cell before transmitting their control packets.

B. Trickle Algorithm

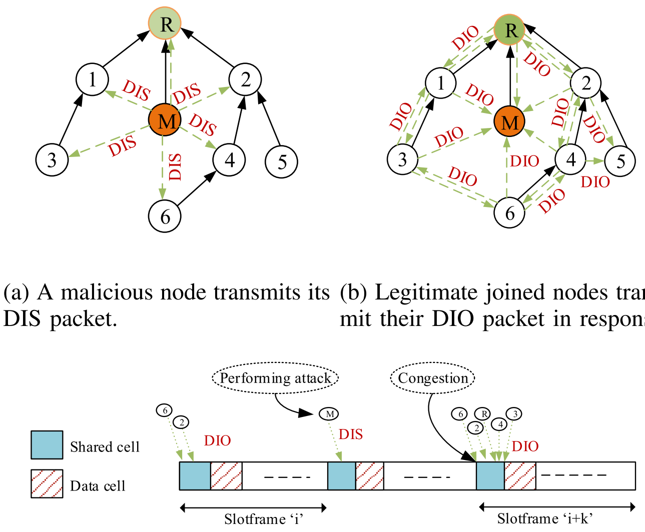
The Trickle algorithm controls the generation and transmission rate of DIO packets to save network bandwidth and nodes' energy. Many parameters are used in Trickle algorithm to generate and transmit DIO packets. The used parameters along with different Trickle steps are described as follows.

- The Trickle algorithm starts with *minimum DIO generation interval* I_{min}.
- The generation interval I is doubled after each interval, until the interval reaches the *maximum DIO generation interval* I_{max} in a steady network.
- In every interval I, each node selects a random time $t \in [\frac{I}{2}, I]$.
- From the interval beginning to t, the node maintains a *counter* c to keep track of the number of consistent DIO packets transmitted by other neighbor nodes. This period is called the *listen-only* period of the node.
- During the listen-only period, if the counter c surpluses the value of predefined (and fixed for all nodes) *redundancy constant* k, then the node suppresses the generated DIO packet. Otherwise, it is transmitted. The value c is reset to 0 after each interval.
- In case of events such as inconsistency in the network, reception of multicast DIS packet, and DODAG version changed, I is reset to the minimum interval I_{min}.

Note that the network administrator sets the values of I_{min}, I_{max} and k at the beginning and keeps unchanged. As previously mentioned, DIO packets are transmitted in shared cells in 6TiSCH networks. Therefore, a node needs to wait for a shared cell after the listen-only period to transmit its generated DIO packet.

III. DIS Attack in 6TiSCH Network

The main goal of a *DIS attack* is to increase the number of transmission of DIO packets in the network, resulting in increased energy consumption by the joined nodes as well as congestion in the shared cell. A congested shared cell will lead to further delay in the transmission of the required control packets, and thus degrades the performance of the network. A malicious user could exploit one of the the Trickle algorithm resetting conditions to increase the congestion level in the shared cell. A node transmits DIS packets when it does not receive DIO packets for some configurable period of time during/after its joining process. Upon receiving a DIS packet, the already joined node resets the DIO generation interval I to I_{min}. This leads to frequent and bursty transmission of DIO packets in the network. In brief, when a legitimate joined node receives a DIS request packet, it assumes that a pledge is probing for network configuration to join the network. Note that a node resets I only when it receives *multicast* DIS request packets. Otherwise, it does not reset the Trickle algorithm, as the node needs to transmit only one DIO packet to the DIS requesting node. If a malicious node intentionally starts transmitting its multicast DIS packets after a certain pre-configured time interval, the receiving legitimates joined nodes reset their Trickle algorithm. This results in frequent transmission of DIO packets (by the DIS receiving joined node) in the network assuming that either the network is inconsistent or a pledge wants to join it. Furthermore, when a node receives unicast DIS packets, it immediately transmits its DIO packet without following the DIO suppression mechanism of the Trickle algorithm. This results in quick energy draining of the DIO transmitting nodes. Furthermore, in 6TiSCH network, both DIS and DIO packets are transmitted in the shared cell of a slotframe. Therefore, when the nodes reset their Trickle algorithm and start transmitting DIO packets frequently, it results in the congestion of the shared cell. Such high congestion in shared cell has a significant impact on the joining time of the pledges [11]–[14]. This DIS attack becomes very difficult to detect when the malicious node transmits DIS packets with modified fictitious identities (i.e., fake `EUI64` address) [7]. In this scenario, the detection algorithms provided



(a) A malicious node transmits its DIS packet.

(b) Legitimate joined nodes transmit their DIO packet in response.

(c) Effect of DIS attack on shared cell's congestion.

Fig. 1. DIS attacks and its affect on 6TiSCH network.

in [8] and [9] do not work effectively as solutions, as they are based on nodes' unique `EUI64` addresses.

Fig. 1 shows the scenario of DIS attack considering a 6TiSCH network consisting of 8 nodes, including the root node Ⓡ and the malicious node Ⓜ. In Fig. 1(a), the malicious node Ⓜ multicasts DIS request packets in the network. In Fig. 1(b), to give response to that DIS request, the neighboring legitimate joined nodes reset their Trickle algorithm and start transmitting their DIO packets in the shared cells. This creates a flooding of DIO packets in the network. Furthermore, as the nodes reset their Trickle algorithm at the same time, so all the nodes participate in the contention race together to transmit their generated DIO packet, which creates congestion in the network. Fig. 1(c) shows the congestion in shared cell before and after the DIS attack. This congestion in shared cell forces the pledges to wait for more time to receive the control packets needed to join the network. Hence, it increases the pledges' joining time and their energy consumption, due to longer channel scanning time. The already joined node also consumes more energy while transmitting DIO packets unnecessarily.

IV. Experimental Evaluation

To analyze the effect of *DIS attack* in RPL based 6TiSCH network during its formation, we design a set of testbed experiments. We use the open-source `Contiki-NG`[2] operating system to generate the binary executable files, which is flashed in Texas Instruments (TI)'s `CC2650 SensorTag` and `CC2650 Launchpad` devices. `Contiki-NG` OS, `CC2650 SensorTag`, and `CC2650 Launchpad` devices support the 6TiSCH communication architecture. In the experimental setup, we program a `CC2650 Launchpad` device as JRC or RPL root node and another `CC2650 Launchpad` device as a malicious node while leaving other nodes as legitimate 6TiSCH nodes. In the experiments, we consider a slotframe size equal to 100 $timeslots$, where each timeslot's duration is 10 ms. We allow each node to generate its EB frame after every 4 s and the legitimate nodes to transmit their DIO packets following the Trickle Algorithm. We use a total number of 16 channels in the network. The JRC starts forming the network and the pledges join the network one by one. The malicious node also joins the network as legitimate node, and is programmed to transmit a periodic multicast DIS request every 60 s.

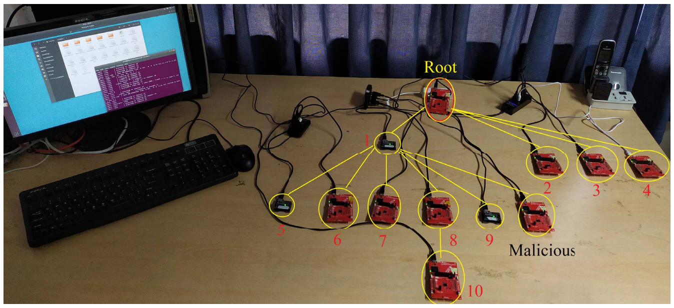


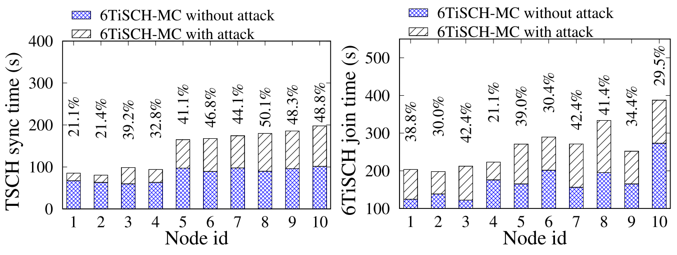
Fig. 2. Experimental testbed.

We consider the following four metrics to validate the effects of a DIS attack to the 6TiSCH network formation. The *TSCH synchronization time* of a pledge is the time needed to become a TSCH synchronized node after receiving a valid EB frame. The *6TiSCH joining time* is the time needed for a node to become a 6TiSCH joined node. This time is important because a node can only transmit its control packets for further expansion of the network after becoming a 6TiSCH joined node. The *number of parent changes* represents the amount of network stability, as parents are changed when a node does not receive any control packet. Lastly, we consider the *energy consumption* of the nodes during the formation of 6TiSCH network. The resource-constrained IoT devices get limited power supply. Hence, the energy consumption of the nodes is of major importance for longer network lifetime.

We run the experiments both in the presence of a malicious node and without the malicious node following the resource allocation policy of 6TiSCH-MC standard. We run each set of experiments 10 times, and results show the average of the obtained results. All figures also show, on top of each bar, the percentage of increase of the considered metric due to DIS attack. Note that at the beginning, we allow the malicious node to join as a pledge to get the synchronization details of the network. Therefore, the malicious node does not perform the DIS attack before getting synchronized (i.e., before receiving an EB frame) with the network.

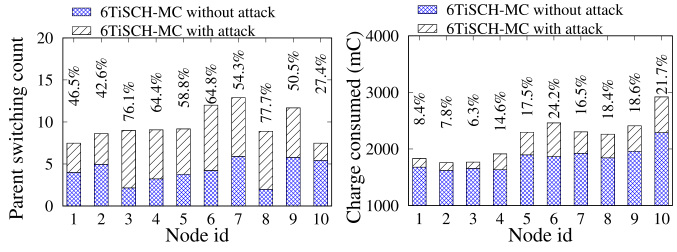
Fig. 3a shows the TSCH synchronization time of the nodes with and without the malicious node. The malicious node, after receiving an EB frame, starts performing DIS attack every 60 s. As previously mentioned, after receiving multicast DIS request, the joined nodes reset their Trickle algorithm and start transmitting mulitcast DIO packet in shared cells, resulting in congestion and packets collisions. Instead, without the malicious node, congestion is less compared to that obtained in the presence of malicious node. So, the TSCH

[2] https://www.contiki-ng.org/



(a) TSCH synchronization time. (b) 6TiSCH joining time.

Fig. 3. Testbed experimental results on average joining times.



(a) Number of parent changes. (b) Charge consumption of nodes.

Fig. 4. Testbed experimental results on network stability and energy consumption by the nodes.

synchronization time is smaller without the malicious node. Fig. 3b shows a similar behavior on 6TiSCH joining time of the nodes. The excessive and unnecessary transmission of the DIO packets affects the transmission of other control packets, which significantly increases the 6TiSCH joining time of the nodes. Furthermore, a node is allowed to transmit its control packets only after completely joining the 6TiSCH network. Therefore, the nodes present at the multihop distance from the root node (i.e., JRC) need to wait more time to complete their parents' joining process.

Fig. 4a shows the number of parent changed by each legitimate node during the experiments. We observe that the legitimate nodes neither can efficiently transmit their control packets (such as unicast DIS), nor they receive other control packets (such as EB, DIO) for long time due to congestion in the shared cell because of DIS attack. Hence, the nodes get de-synchronized with the TSCH network and try to re-join as fresh pledges, which increases the parent switching count. Note that after re-joining the network, the value of parent switching count of the node is increased by one even if it joins the same parent. This increased number of parent changes affects the network stability and increases the control packet overhead in the network, which further increases the energy consumption of the nodes.

Fig. 4b shows the average charge consumption of the nodes during the experiments which is directly proportional to the energy consumption of the nodes. We calculated the charge consumption of the nodes from their *radio duty cycles* obtained from the Contiki-NG's "*energest*" module as described in the work in [13]. The already joined nodes unnecessarily transmit DIO packets in the network due to the DIS attack, increasing the energy consumption of the DIO transmitting joined nodes by approximately 16%. On the other hand, the malicious node consumes an average $2245.9\ mC$ in one hour, which is less/similar compared to that in the legitimate nodes. This increasing energy consumption of the nodes shortens the network lifetime, thus requiring frequent battery charging sessions or node replacements.

V. Conclusion and Future Work

In this work, we showed the impact of DIS attack on 6TiSCH network formation. From the experiments, it can be seen that the DIS attack significantly increases the synchronization and joining time of pledges. It also increases the energy consumption of the nodes, and so significantly impacts on the network's lifetime. We showed that the malicious node does not require a large amount of energy to create impact on the network performance. Furthermore, the malicious node does not need sensitive information (e.g., cryptographic keys) to perform the DIS attack. It just needs to get synchronized with the network, specifically, to know the timeslot and channel where the joined nodes transmit their control packets. In the future, we plan to build a dataset by performing different routing attacks in a real IoT network and use an efficient machine learning technique to detect the attacks.

References

[1] *IEEE Standard for Low-Rate Wireless Networks*, Standard 802.15.4-2015 Std 802.15.4-2011, Apr. 2016, pp. 1–709.

[2] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 595–615, 1st Quart., 2020.

[3] X. Vilajosana, K. Pister, and T. Watteyne, *Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration*, document RFC 8180, Internet Engineering Task Force, May 2017.

[4] T. Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, Internet Engineering Task Force, Mar. 2012.

[5] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing attacks and mitigation methods for RPL-based Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1582–1606, 2nd Quart., 2019.

[6] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO suppression attack against routing in the Internet of Things," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2524–2527, Nov. 2017.

[7] C. Pu, "Spam DIS attack against routing protocol in the Internet of Things," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 73–77.

[8] B. Farzaneh, M. A. Montazeri, and S. Jamali, "An anomaly-based IDS for detecting attacks in RPL-based Internet of Things," in *Proc. 5th Int. Conf. Res. (ICWR)*, Apr. 2019, pp. 61–66.

[9] A. Verma and V. Ranga, "Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, p. e3802, Feb. 2020.

[10] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks–Specific Requirements—Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPAN)*, Standard 802.15.1-2005 802.15.1-2002, pp. 1–700, 2005.

[11] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving network formation in 6TiSCH networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 98–110, Jan. 2019.

[12] A. Kalita and M. Khatua, "Channel condition based dynamic beacon interval for faster formation of 6TiSCH network," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2326–2337, Jul. 2021.

[13] A. Kalita and M. Khatua, "Opportunistic transmission of control packets for faster formation of 6TiSCH network," *ACM Trans. Internet Things*, vol. 2, no. 1, pp. 1–29, Jan. 2021.

[14] A. Kalita and M. Khatua, "Adaptive control packet broadcasting scheme for faster 6TiSCH network bootstrapping," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17395–17402, Dec. 2021.