

# Broadcasting strategies in 6TiSCH networks

Mališa Vučinić<sup>1</sup> | Thomas Watteyne<sup>1</sup> | Xavier Vilajosana<sup>2</sup>

<sup>1</sup>EVA team, Inria, Paris, France

<sup>2</sup>Wireless Networks Research Lab, Universitat Oberta de Catalunya, Barcelona, Spain

## Correspondence

Mališa Vučinić, Inria, 2 rue Simone Iff, 75012 Paris, France.

Email: mališa.vucinic@inria.fr

## Funding Information

FEDER regional development fund, TEC2015-71303-R. Spanish Ministry of Economy. European Commission, H2020 F-Interop, H2020 ARMOUR.

We consider IPv6-enabled networks that run on top of the time-slotted channel hopping mode of IEEE802.15.4 (6TiSCH). The ongoing discussions in the standardization community concern the network formation process and the definition of a bootstrapping protocol by which a new mote is admitted into the network. Because the bootstrapping traffic uses the same shared slots as the network broadcasts, the key to the optimal performance of the network formation process lays in the optimization of the network broadcasting strategy. The problem boils down to the issue of stabilizing slotted Aloha. To do so, we adapt a broadcast algorithm to the specifics of 6TiSCH networks. By simulation, we evaluate the optimal broadcast transmission probability in the network. We answer the open questions in the IETF 6TiSCH standardization community that concern the performance of the network formation process for the optimal values of transmission probability. As the main contribution of the letter, we provide network administrators with a set of values that allow the formation of dense networks.

## KEYWORDS

6TiSCH, broadcast, IEEE 802.15.4, networks, performance analysis

## 1 | INTRODUCTION

6TiSCH is a technology being standardized by the Internet Engineering Task Force (IETF) which combines the performance of IEEE802.15.4 Time-slotted Channel Hopping (TSCH) with the interoperability of IPv6. 6TiSCH is a key enabling low-power wireless technology for the Industrial Internet of Things (IIoT). The working group currently focuses most of its effort on the bootstrapping protocol that allows a new mote to be admitted into the network. This protocol is currently under heavy discussion within the security design team at the IETF 6TiSCH working group; this letter addresses the open questions that concern its performance.

In 6TiSCH networks, the bootstrapping traffic is combined with broadcast traffic and transmitted in shared slots in a slotted Aloha fashion. For this reason, the performance of the bootstrapping protocol is inherently dependent on the broadcasting strategy used in the network. This letter proposes an optimization of the performance of a generic bootstrapping protocol by revisiting the broadcasting strategy in the network.

As each mote joins the network, it starts broadcasting within a single shared slot that is defined by the recently standardized “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration.”<sup>1</sup> Broadcasts in 6TiSCH networks include Enhanced Beacons (EBs)—used by new motes to synchronize to the network—and RPL DIOs—used by motes to build a directed acyclic graph for routing purposes. In dense networks, this creates congestion: broadcast traffic of different motes collides due to the limited bandwidth. Effectively, this limits the formation of large networks when the instability point of slotted Aloha is reached.

State-of-the-art approaches<sup>(2–4)</sup> address the problem by assigning additional bandwidth; this comes with extra maintenance and energy cost.<sup>5</sup> Using the Trickle timer<sup>6</sup> to schedule EBs is not feasible in 6TiSCH as new motes have no means of soliciting its reset without being synchronized with the network. We adopt a different approach and adapt the Bayesian broadcast algorithm (7) to stabilize slotted Aloha. In Bayesian broadcast, each broadcast frame is transmitted with a given probability of transmission. It is possible to maximize the probability that exactly one transmission occurs during the slot by using an optimal value of

probability of transmission. However, in the 6TiSCH case, apart from the broadcast traffic, bootstrapping traffic also ends up in the slotted Aloha slot. Depending on the load of the bootstrapping traffic, the optimal probability of transmission changes.

We simulate 6TiSCH networks for different bootstrapping loads to evaluate the optimal probability of transmission. Then, to answer the open questions in the IETF 6TiSCH working group, we evaluate the performance of the bootstrapping protocol with these optimal values of broadcast probability of transmission. The contributions of this letter are 3-fold:

- We provide a practical adaptation of the Bayesian broadcast algorithm to 6TiSCH networks.
- We extend the 6TiSCH simulator<sup>1</sup> to simulate the shared slots, downward and bootstrapping traffic and provide the open-source extension for the benefit of the community.
- We provide extensive results on the duration of the network formation process in 6TiSCH networks, as well as on the optimal set of parameters to run a 6TiSCH network.

The remainder of this letter is organized as follows. Section 2 outlines the network formation process in 6TiSCH networks. We adapt the Bayesian broadcast algorithm to 6TiSCH in Section 3. Section 4 studies the performance of the proposed approach and gives estimates on the duration of the network formation process. Section 5 concludes this letter.

## 2 | NETWORK FORMATION IN 6TISCH NETWORKS

In a 6TiSCH network, all motes periodically transmit EB frames. Motes that are joining the network continuously listen for EBs sent by nearby motes. The EB contains enough information for a new mote to synchronize to the slotframe and the slot boundary of the given network, once an EB is received. Because transmissions in a TSCH network cycle over all available channels, to hear an EB, a new mote needs to keep its radio constantly listening on a single channel. A well-known approach is that motes select a random channel to listen on, which distributes the load on the network by new motes joining. This effectively divides the load by the number of channels, as motes receive EBs at different instants and therefore independently trigger the join process.

The reception of an EB indicates to the mote the position of a shared cell that is used for network bootstrapping.<sup>1</sup> All broadcast traffic in the network—as well as the bootstrapping (join) unicast traffic—is transmitted within this shared cell, in a slotted Aloha fashion. Bootstrapping (join) traffic refers to frames exchanged as a result of the security protocol by which a new mote is admitted into the network. We consider a generic bootstrapping protocol with variable number of round trips in order to shed light on how this protocol design consideration affects the network performance. The broadcast traffic is divided into EBs and the routing protocol overhead, which in 6TiSCH are RPL DIO messages.<sup>8</sup> DIOs are used for the construction of a directed acyclic graph for upward routing. As each mote joins, it starts emitting EBs and DIOs in the slotted Aloha cell.

Once first-hop motes join and start transmitting EBs, they extend the range of the network by serving as relays to the second-hop motes. The process repeats, with the critical part being the radio access in the slotted Aloha manner. The network formation finalizes when the last-hop motes join.

The problem arises as more and more motes join, they start congesting the slotted Aloha cell with EBs and DIOs, preventing new motes to join. Related work tackles the problem by assigning additional cells for maintenance traffic.<sup>(2–4)</sup> Such approaches—which statically assign a set of additional cells—incur additional energy consumption and management overhead. They are also not compliant to the recently standardized “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration.”<sup>1</sup>

## 3 | BAYESIAN BROADCAST ALGORITHM IN 6TISCH

The shared cell in a 6TiSCH network needs to accommodate three types of traffic<sup>1</sup>: EBs<sup>2</sup>; DIOs<sup>3</sup>; bootstrapping—or “join”—traffic. Our goal is to tune the probability of transmission of these frames in order to lower the contention and so operate at a stable point of slotted Aloha. We therefore propose the adaptation of the Bayesian Broadcast Algorithm by Rivest<sup>7</sup> to 6TiSCH networks.

The basic idea of the Bayesian Broadcast Algorithm is simple: at each shared cell, each mote in a network transmits a (broadcast) packet with a probability  $p_{tx}$ . Given a fully-meshed network with  $N$  motes, the probability that there is exactly one transmission in a shared cell is denoted as  $P_{success}$  and given in Equation (1).

$$P_{success} = N p_{tx} (1 - p_{tx})^{N-1}. \quad (1)$$

The probability of success is highest for  $p_{tx} = 1/N$ .<sup>7</sup> We consider  $N$  as known, as it is easy to estimate locally using, for example, received EBs. The problem in the 6TiSCH case, however, is how to locally configure the broadcasting probability of different types of traffic, taking into account a nonbroadcast component  $p_{join}$  caused by bootstrapping. The  $p_{join}$  component is the result of nondeterministic traffic sent by each mote during bootstrapping stage.

$$p_{tx} = p'_{EB} + p'_{DIO} + p_{join}. \quad (2)$$

We maximize throughput when the value of  $p_{tx}$  in Equation (2) is  $1/N$ . For each slotted Aloha transmission opportunity (ie, for each shared cell), each mote flips a biased coin and sends an EB with a probability  $p_{EB}' = p_{EB}/N$ , or a DIO with a probability  $p_{DIO}' = p_{DIO}/N$ . It is important to stress that  $N$  includes only motes that have already joined the network so it dynamically changes during the formation process. In Section 4, we simulate the network formation process to evaluate the optimal value of  $p_{EB}$  for different load of the bootstrapping protocol. The load of the bootstrapping protocol affects the  $p_{join}$  component in Equation (2).

Equation (1) considers the case where  $N$  motes are present in the network and are all broadcasting DIOs and EBs. During the network formation process, this is not the case as motes join one by one, and only start broadcasting once joined. By conditioning  $N$ , we can approximate Equation (1) in the network formation by Equation (3).

$$P'_{success} = \sum_{n=1}^N n p_{tx} (1 - p_{tx})^{n-1} \times P(N = n). \quad (3)$$

Similarly to Equation (1), the probability of a “hole” (per the terminology of 7, the case where there is no packets transmitted in the slotted Aloha cell) with  $N$  motes broadcasting is expressed in Equation (4).

$$P_{hole} = (1 - p_{tx})^N. \quad (4)$$

In the network formation case, the probability of a hole can be found similarly to Equation (3). And finally, the probability of collision is expressed in Equation (5).

$$P_{collision} = 1 - P_{success} - P_{hole}. \quad (5)$$

A metric of interest in the 6TiSCH case is  $T_{FB}$ , the mean time between the moment a new—unsynchronized—mote is powered on, and the moment it receives a first beacon. Considering the minimal schedule<sup>1</sup> where there is one opportunity to transmit broadcast once per slotframe, this time is expressed in Equation (6).

$$T_{FB} = \frac{L_S \times C}{p_{EB}}. \quad (6)$$

In Equation (6),  $L_S$  denotes the slotframe length and  $C$  number of channels in use.

## 4 | EVALUATION

We want to answer the following question: *what is the optimal configuration of a 6TiSCH network, given the traffic load of the bootstrapping protocol and maximal number of motes in a radio neighborhood?* By simulation, we find the optimal value of  $p_{EB}$ ; given that value, we further study the overall duration of network formation process and its energy consumption.

We extend the 6TiSCH simulator to support shared cells and downward traffic. Our contributions are open-source for the benefit of the community. The simulator implements the bootstrapping schedule according to RFC8180.<sup>1</sup>

We perform the simulations for fully-meshed networks, as this represents the worst case in terms of slotted Aloha contention. For all results in this section, we fix  $p_{DIO}$  to one third of the optimal probability of transmission in order to permit the construction of the directed acyclic graph and give enough room for the bootstrapping traffic and EBs (see Equation (2)). The effect of  $p_{DIO}$  setting can only be seen in multihop topologies. We found one third to be a good compromise between the traffic load and the speed of convergence of the directed acyclic graph.

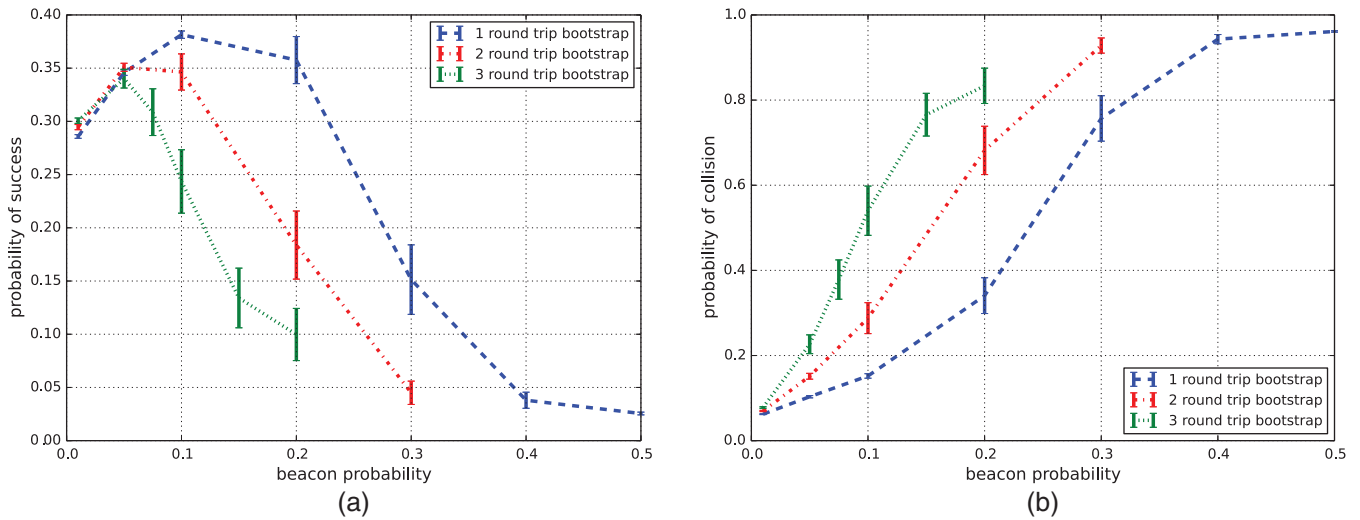
We measure the probability of success, that is, the probability that exactly one transmission occurs during the shared slot, as it gives us an idea how throughput behaves as a function of traffic load. We also measure the probability of collision, that is, the probability that more than one transmission occurs during the shared slot, in order to understand the relation of traffic load and wasted transmissions. Finally, we study the performance of the overall join process in terms of its duration and energy consumption as it gives us an insight on how costly it is to build a network at startup.

We fix the probability by which DIOs are sent and vary the traffic load by increasing the value of  $p_{EB}$ . We also vary the traffic load of the bootstrapping protocol by increasing the number of required round trips. We expect that, with the increased bootstrapping load, the optimal value of  $p_{EB}$  decreases due to the increased load on the shared cell.

Each data point is an average of 100 simulation runs; we plot 95% confidence intervals.

The probability of success increases for small values of beacon probability (see Figure 1A) due to the fact that there is not enough traffic generated in the network and the shared cell remains unoccupied. It hits the optimal value for beacon probability that is around 0.1 for 1 and 2 round-trip bootstrap protocol. The performance degrades after the optimal peak as there is increasing load on the slotted Aloha cell and contention appears. Because probability of success corresponds to the normalized throughput, we can see that the higher load of the bootstrapping protocol effectively reduces the network throughput during the join process.

The reduced throughput with increasing bootstrapping load is the consequence of a higher number of collisions, which can be seen in Figure 1B. Note that, for the optimal value of beacon probability (0.1), the pass from a single round-trip bootstrap



**FIGURE 1** A. Probability of success for a 40-mote fully-meshed network and  $p_{DIO} = 1/3$ . The increased bootstrapping load reduces the throughput during network formation process due to the increase in collisions. B. Probability of collision for a 40-mote fully-meshed network and  $p_{DIO} = 1/3$ . For optimal  $p_{EB} = 0.1$ , collisions double with increasing bootstrapping load

protocol to two round trips doubles the collisions. As expected, probability of collision increases with beacon probability, as the load on the slotted Aloha cell increases.

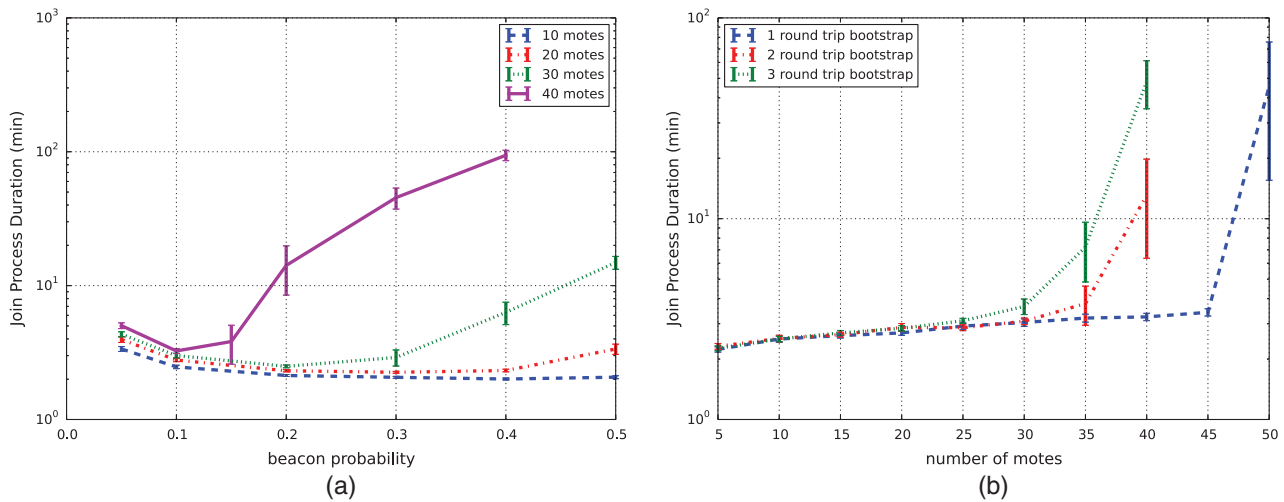
The duration of the join process represents the time until the last mote in the network has joined, that is, has performed all exchanges of the join protocol. It is an important metric as it tells us from the practical standpoint how much slotted Aloha contention affects the network setup time. We consider the worst case where all motes are switched on at the same time, the root being switched on last. The duration of the join process takes into account the initial delay while the motes are unsynchronized and are constantly receiving, waiting for EBs. Figure 2A shows the results for a single round-trip protocol. For smaller values of  $p_{EB}$ , the time until first beacon decreases, and with it the duration of the join. For beacon probabilities larger than  $p_{EB} = 0.1$ , the dominant component is the time it takes for the mote to join the network after it has received the beacon. As for larger beacon probabilities, the load on the slotted Aloha cell is larger, so is the probability of collision (see Figure 1B) which prolongs the time it takes for motes to perform the exchanges of the join protocol. Figure 2B shows the duration as a function of the number of motes in the fully-meshed network. We can see that the join duration “explodes” after a given number of motes in the network are reached which effectively prevents the construction of larger networks. Therefore, in order for large networks to form within a reasonable amount of time, it is necessary to reduce the beacon probability and so increase “room” for the  $p_{join}$  component in Equation (2).

We proceed by estimating the average charge consumed per mote in Figure 3 for the LTC5800 chip from Linear Technology based on the values reported in its datasheet.<sup>2</sup> We use this particular chip as it is best-in-class for power consumption, as is commercially available. We can see that the trend follows Figure 2A, although the optimal value is shifted to the right. This means that the optimal value of  $p_{EB}$  in terms of the overall join duration is not the same when it comes to energy consumption. The reason is that motes spend more energy while initially unsynchronized and constantly listening for beacons. Therefore, the energy consumption benefits higher  $p_{EB}$ , as the time until first beacon is shorter (see Equation (6)). To put the numbers in Figure 3 into perspective, consider that an alkaline AA battery typically carries 7920 C (2200 mAh), so that the charge consumed during the join process for  $p_{EB} = 0.1$  represents around 0.001% of the total charge. For this System on Chip (SoC), the average current consumption during the join is around  $500 \mu\text{A}$ , which represents a factor of 5 to 50 increases in comparison to the typical consumption during regular network operation ( $10\text{--}100 \mu\text{A}$  average typical). Different boards have, however, different consumption. Based on the measured values presented in Vilajosana et al.,<sup>5</sup> in Table 1 we compare the total charge consumed during the join process for  $p_{EB} = 0.1$  and  $p_{DIO} = 1/3$  for LTC5800 SoC, OpenMote-STM32F board and GINA.

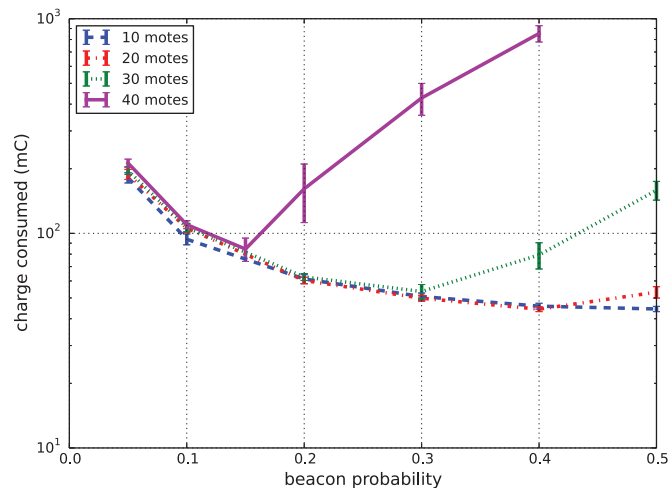
Finally, in Figure 4 we compare in terms of collision probability the Bayesian approach with periodic beaconing, current state-of-practice in open-source implementations of TSCH. As expected, the periodic approach results in exponential increase in collision probability with increasing number of motes in the network. Depending on the EB and DIO periods used in the network, the performance lines cross for approximately 13, 19, and 25 motes in the fully-meshed network. The advantage of Bayesian broadcast can clearly be seen as the network density increases.

## 5 | CONCLUSION

This letter presents an adaptation of the Bayesian broadcast algorithm to optimize the network formation process in 6TiSCH networks. Through extensive simulation, we parametrize the network and stabilize the slotted Aloha cell in the standards-compliant



**FIGURE 2** A. Join duration for a single round-trip bootstrap protocol and  $p_{DIO} = 1/3$ . For smaller values of  $p_{EB}$ , the dominant component of the duration is the time the mote spends unsynchronized waiting for EBs. For larger values (and increased traffic load) bootstrap retransmissions prevail. B. Join duration for beacon probability  $p_{EB} = 0.1$  and  $p_{DIO} = 1/3$ . For a given join process duration, the number of motes in the network decreases with increasing traffic load



**FIGURE 3** Charge consumed during the network formation assuming a LTC5800 System-on-Chip is used. The value of  $p_{EB}$  that minimizes energy consumption is slightly higher compared to Figure 2a due to the initial unsynchronized period where motes constantly listen for EBs

**TABLE 1** Charge consumed during join in a 40-mote network with different hardware

Board	Total charge [mC]
LTC5800	109.8
GINA	380.9
OpenMote-STM32F	905.6

minimal profile defined by the IETF 6TiSCH working group. For networks with up to 40 motes in a radio neighborhood, we find that setting  $p_{DIO}$  to one third and  $p_{EB}$  to 0.1 gives good results in terms of network formation time. The load of the bootstrapping protocol affects the maximal number of motes that can be in a radio neighborhood. We found that with a single round-trip bootstrap protocol, we can build networks that have up to 45 motes in a radio neighborhood. With 2 and 3 round-trip protocols, this number decreases to 35 and 30, respectively.

**NOTES**

1. <https://bitbucket.org/6tisch/simulator>.
2. [www.linear.com/docs/41870](http://www.linear.com/docs/41870).

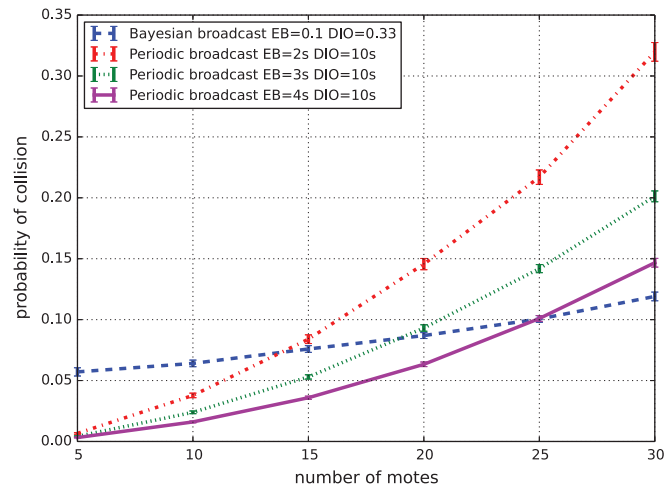


FIGURE 4 Comparison of Bayesian and periodic broadcast techniques

#### ACKNOWLEDGMENTS

This work was supported by European Commission through the H2020 ARMOUR and H2020 F-Interop projects, and by the Spanish Ministry of Economy and the FEDER regional development fund under SINERGIA project (TEC2015-71303-R).

#### ORCID

Mališa Vučinić  <http://orcid.org/0000-0002-7700-9121>

Thomas Watteyne  <http://orcid.org/0000-0002-3695-9315>

Xavier Vilajosana  <http://orcid.org/0000-0002-3020-427X>

#### REFERENCES

- Vilajosana X, Pister K, Watteyne T. Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) configuration. RFC 8180; 2017.
- Duquennoy S, Al Nahas B, Landsiedel O, Watteyne T. Orchestra: robust mesh networks through autonomously scheduled TSCH. Paper presented at: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys):337–350ACM; 2015; New York, NY.
- Khoufi I, Minet P, Livolant E, Rmili B. Building an IEEE 802.15. 4e TSCH network. Paper presented at: 35th IEEE–International Performance Computing and Communications Conference, Las Vegas, NV, USA; 2016.
- Vogli E, Ribezzo G, Grieco LA, Boggia G. Fast join and synchronization schema in the IEEE 802.15. 4e MAC. Paper presented at: Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE:85–90IEEE, New Orleans, LA, USA; 2015.
- Vilajosana X, Wang Q, Chraim F, Watteyne T, Chang T, Pister K. A realistic energy consumption model for TSCH networks. *IEEE Sens J.* 2014;14(2):482–489.
- Levis P, Patel N, Culler D, Shenker S. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. Paper presented at: Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation, San Francisco, CA, USA; 2004.
- Rivest R. Network control by Bayesian broadcast. *IEEE Trans Inform Theory.* 1987;33(3):323–328.
- Winter T, Thubert P, Brandt A, et al. RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550; 2012.

**How to cite this article:** Vučinić M, Watteyne T, Vilajosana X. Broadcasting strategies in 6TiSCH networks. *Internet Technology Letters.* 2018;1:e15. <https://doi.org/10.1001/itl2.15>