In-Band Network Telemetry in Industrial Wireless Sensor Networks

Abdulkadir Karaagac, Eli De Poorter, Jeroen Hoebeke

***Abstract*—With the emergence of the Internet of Things (IoT) and Industry 4.0 concepts, industrial applications are going through a tremendous change that is imposing increasingly diverse and demanding network dynamics and requirements with a wider and more fine-grained scale. Therefore, there is a growing need for more flexible and reconfigurable industrial networking solutions complemented with powerful monitoring and management functionalities. In this sense, this paper presents a novel efficient network monitoring and telemetry solution for Industrial Wireless Sensor Networks mainly focusing on the 6TiSCH Network stack, a complete protocol stack for ultra-reliable ultra-low-power wireless mesh networks. The proposed monitoring solution creates a flexible and powerful in-band network telemetry design with minimized resource consumption and communication overhead while supporting a wide range of monitoring operations and strategies for dealing with various network scenarios and use cases. Besides, the technical capabilities and characteristics of the proposed solution are evaluated via a real-life implementation, practical and theoretical analysis. These experiments demonstrate that in-band telemetry can provide ultra-efficient network monitoring operations without any effect on the network behavior and performance, validating its suitability for Industrial Wireless Sensor Networks.**

***Index Terms*—Industrial Internet of Things (IIoT), In-Band Network Telemetry (INT), IEEE 802.15.4e TSCH, 6TiSCH, Network Monitoring, Remote Network Management.**

I. Introduction

Over the past decade, we have seen the emergence of several industrial wireless sensor networking technologies based on a Time-Slotted Channel-Hopping (TSCH) scheme to meet the stringent requirements of industrial applications, including WirelessHart [1] and ISA100.11a [2], and finally 802.15.4e TSCH [3] and 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4e) [4]. Compared to its TSCH-based predecessors, 6TiSCH has opted for an open and standardized communication stack as well as support for different scheduling schemes, turning it into a more open and flexible, but also equally reliable and deterministic wireless communication solution.

However, for continuous, persistent and problem-free operation of these Industrial Wireless Sensor Networks (IWSN), regardless of the technology, it is critical to have visibility and awareness into what is happening on the network at any one time. On the one hand, these technologies are used in numerous use cases with strong needs for deterministic and real-time network services with latency and reliability guarantees [5], where timely handling of critical events is crucial so that system failures can be avoided or at least system downtime can be minimized [6]. On the other hand, these networks are generally subject to a variety of environmental factors, harsh conditions and challenges throughout their lifetime. Therefore, it is vital for these networks to monitor network devices continuously or periodically in order to guarantee their functioning, detect and isolate relevant problems and verify if all system requirements are being met simultaneously.

Nevertheless, efficient monitoring and management mechanisms for these networks have not been addressed adequately in the existing literature. Firstly, traditional probing or polling-based network monitoring approaches for IP networks are of limited applicability in these constrained and dynamic networks due to their static and inefficient design. Especially, considering the dynamic nature of sensor networks, the introduced control traffic can occupy extensive network resources, impact network behavior and/or interfere with the scheduled application traffic flow. Secondly, there are also a number of efficient health monitoring and debugging methods available for WSNs [7]–[10], which can only offer limited capabilities for collecting in-network state information and telemetry data. Therefore these techniques are not sufficient for advanced network monitoring and management operations such as traffic engineering, network optimization and anomaly detection.

There is a need for network monitoring mechanisms for IWSNs that consider the specific characteristics and requirements of these networks. Firstly, network resources are limited and link layer frames are subject to strict size constraints. So, the monitoring solution needs to be light-weight to reduce its resource consumption and minimize communication overhead. Secondly, it needs to be adaptive to frequently changing network conditions and dynamic topologies. Thirdly, it should be robust to fight hostile wireless channel effects or temporary/permanent node failures, so not relying on one or a subset of specific network nodes. Lastly, it needs to support a wide range of monitoring operations with extensive and useful detection capabilities coupled with dedicated analysis systems to collect, trend and correlate observed activity.

In this regard, the key research problem of this work is the design of more intelligent and efficient monitoring solutions and flexible management architectures for wireless industrial networking technologies, more specifically the 6TiSCH Network stack [4]. For this purpose, the recently proposed concept of In-Band Network Telemetry (INT), by the P4 Language

Manuscript received XXXX XX, 2019; revised XXXXXXX XX, 2019. This work was partially funded by the FWO-Flanders, under grant agreement #G055619N. Part of this research was funded by the ICON project Internet of Shipping (IoS). IoS is realized in collaboration with imec, with project support from VLAIO. Project partners are imec, Exmar, Ovinto and Aloxy.

The authors are with the IDLab, Department of Information Technology, Ghent University–imec, 9052 Gent, Belgium (e-mail: abdulkadir.karaagac@ugent.be; eli.depoorter@ugent.be; jeroen.hoebeke@ugent.be).

Consortium [11], is adopted, which allows the collection and reporting of network state without any need for artificial probing packets and at the exact moment when real user traffic traverses them [12]. Since the state-of-the-art INT techniques are only being explored for wired networks [11], these mechanisms need to be redesigned in order to meet the aforementioned key requirements.

The proposed monitoring telemetry solution creates an efficient, adaptive and flexible design which offers several novel monitoring functionalities and telemetry operations for 6TiSCH Networks. Initially, it creates an opportunistic piggybacking mechanism by exploiting the remaining space in the 802.15.4e frames without any effect on the application traffic itself. Therefore, it eliminates the need for any resource reservation for monitoring data in IWSNs. Secondly, it enables the nodes to follow different INT initiation, addition and encoding strategies. For this, it creates a self-organizing and distributed telemetry solution by enabling middle nodes to initiate INT or to decide what to add or skip as telemetry. Thirdly, the proposed solution also allows downlink in-band telemetry or query and polling-based monitoring operations if needed. Thanks to all these novel functionalities, the proposed solution enables efficient monitoring systems for IWSNs, which can be used to monitor network performance, troubleshoot and isolate problems, verify services and perform traffic engineering (i.e. scheduling, routing) and network optimization.

The main contributions of this paper can be summarized as follows:

- The first, to the best of found knowledge, conceptual design for an In-Band Network Telemetry (INT) mechanism adapted to Wireless Sensor Networks.
- A flexible and adaptive network monitoring mechanism, with minimal overhead, for IEEE 802.15.4e TSCH Networks, which was inspired by INT.
- The definition of telemetry semantics and data models for 6TiSCH Networks and their efficient encoding in the IEEE 802.15.4 frames.
- The definition of novel telemetry operations and strategies for dealing with various network scenarios and system interactions.
- The implementation of the proposed design and end-to-end validation and evaluation of the proposed architecture via a real-life 6TiSCH Network with the ability of INT.
- An analysis of the technical capabilities and characteristics of the proposed solution, validating its suitability for handling various monitoring applications and scenarios.

The remainder of the paper is organized as follows. Section II introduces relevant concepts and provides a brief overview of the related work. After that, the proposed INT-based capacity-neutral network monitoring mechanism for IWSNs is presented in Section III. Section IV presents more details about the solution along with the practical implementation. Next, Section V presents an evaluation and validation study about the proposed solution via theoretical and practical experiments, which is followed by a detailed discussion about the application potential and novel monitoring capabilities in Section VI. Finally, Section VII concludes the paper.

II. Technical Background and Related Work

A. IEEE 802.15.4e TSCH and 6TiSCH

IEEE 802.15.4e is a recent MAC amendment of the IEEE 802.15.4 standard, specially designed for harsh industrial environments with a reliable and deterministic communication scheme based on Time-Slotted Channel Hopping (TSCH) [3]. In a TSCH network, time is sliced up into time slots and the overall communication is orchestrated by a schedule which defines the action (transmit, receive, sleep) of each node in each time slot [3]. The proper functioning of a TSCH network depends on this schedule which can be typically created in various ways, but should be computed according to the specific requirements of the applications, such as latency, reliability and energy.

Recently, a new IETF Working Group (WG), named 6TiSCH, has been formed to investigate IPv6 connectivity over the TSCH mode of the IEEE 802.15.4e protocol [4]. The WG has defined an operation sub-layer (6top) in order to bind the prior IPv6-enabled standards (6LoWPAN, RPL, and CoAP) with IEEE 802.15.4e TSCH. It targets to create a standardized approach to build and maintain a schedule, perform TSCH configuration and control procedures. The 6TiSCH protocol defines four approaches to manage the TSCH schedule: static (i.e. shared cells), neighbor-to-neighbor (soft cells), remote (hard cells) and hop-by-hop scheduling [4]. The Neighbor-to-Neighbor scheme enables motes to negotiate and agree on a schedule by using distributed scheduling protocols, whereas in remote schedule management, a central entity, called Path Computation Element (PCE), is continuously adjusting the TSCH schedule according to the network state and traffic requirements. An overview of the 6TiSCH Network Architecture and scheduling schemes are presented in Figure 1.

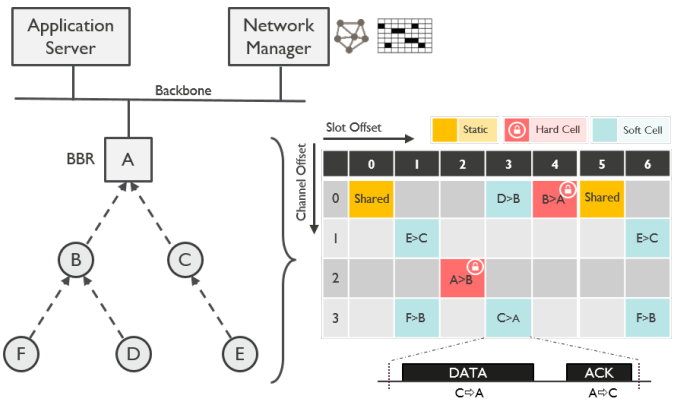


Fig. 1: Architecture and scheduling in 6TiSCH Networks.

As described in [13], 6TiSCH defines various network and schedule management schemes in a centralized, distributed and mixed fashion. While, in WirelessHART and ISA100.11a, the central network manager regulates all the communication in the network. Regardless of the technology, any of the centralized management tasks requires collecting and analyzing measurement data, often in real-time, through a process called network telemetry. These measurement data can typically be node or network state information such as health/failure reports, link/neighbor statistics, network topology and node/link

occupancy [14]. Based on this data, the Network Management Entity (NME) can perform a wide range of management operations, including anomaly detection, traffic engineering (schedules, routes), network optimization, service verification.

B. Network Monitoring in WSNs

In traditional networks, fault detection and isolation has been done using tools like *ping* and *traceroute* or more complex systems built on out-of-band management (active probing, path tracing) [15]. More recently, with the emergence of Software-Defined Networking (SDN), network traffic monitoring became an ecologically vital mechanism, [16], and a variety of monitoring technologies were developed to monitor network traffic, including SNMP and NETFLOW [17]. However, all these network monitoring approaches are based on the idea of streaming telemetry, statistical polling or active probing. So, they introduce additional network traffic and may impact the network behavior, and therefore not suitable for any kind of WSN technology. Moreover, these end-to-end tools are generally not sufficient for detecting and reporting transient network congestion and isolating fault location [15].

There are also several monitoring and diagnostic techniques which are specifically designed for WSN technologies [18]. Firstly, there are active health monitoring systems, in which the sensor nodes are actively sending local status resulting in undesired extra monitoring traffic [7], [19]. For instance, in WirelessHart, the network management fully relies on periodical or event-driven information or alarm reports from field devices to Network Manager [14]. In contrast to these active solutions, there are also passive network diagnosis and tomography techniques, which aim to minimize interference and simplify debugging of sensor networks [8], [10], [20], [21]. However, the network sniffing-based passive monitoring solutions use the idea of passively collecting message traces throughout the network, which makes it impractical for continuous network monitoring and management, especially for large networks. On the other side, inference-based passive approaches can only offer limited accuracy and functionality for fine-grained industrial wireless sensor networks.

While, there are also hybrid monitoring solutions which try to combine both active and passive approaches to realize greater observability of the monitored WSNs [22]. For instance, *Keller et al.* [9] proposes a hybrid health monitoring system that utilizes passively reconstructed packet information while only adding one bit extra information to improve the failure detection accuracy. However, this approach assumes that certain network events always result in a measurable additional delay to the end-to-end packet delivery, which is not the case all the time.

In addition, the 6TiSCH WG is defining a management interface, based on CoAP Management Interface (CoMI) [23], which can be used to monitor network performance and perform network configurations. However, performing telemetry via CoMI interfaces will result in a polling-based monitoring scheme which may cause a large amount of control traffic. In [24], a piggybacking mechanism is proposed to individually monitor 6TiSCH network performance, only targeting end-to-end delivery ratio and delay, via dedicated control messages. In [25], the authors present a framework based on a poller-pollee architecture in order to monitor Low Power and Lossy Networks (LLN).

The scope of all these diagnosis and monitoring solutions are quite limited in terms of efficiency, flexibility or capability, considering the required functionalities for advanced network management operations for the IWSNs: such as schedule discovery and monitoring, QoS verification, network awareness, network resource and utilization monitoring, in-depth problem troubleshooting and isolation. They also do not consider several network monitoring scenarios such as downlink and node-to-node communication. Therefore, this work investigates a flexible, efficient and novel network monitoring solution for IWSNs which can be used to realize various simple and also complicated monitoring applications with a minimized cost and zero effect on the network behavior.

C. In-band Network Telemetry

As an alternative to the traditional monitoring techniques, In–Band Network Telemetry (INT) is recently proposed as a framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane [11]. More recently, the INT methodology is also referred to as In-situ Operations, Administration, and Maintenance (iOAM) [26]. INT, or iOAM, is created to complement current out-of-band monitoring (also called "active" telemetry) mechanisms and allows for telemetry metadata to be collected as packets traverse a network [26]. The term "in-band" refers to the fact that telemetry data is carried within data packets rather than being sent within specifically dedicated packets. Therefore, it does not need for artificial probing packets or dedicated middle-boxes, and the network state is obtained at the exact point in time the real user traffic passes through [12]. Also, the insertion of in–band information does not change the forwarding behavior of the packet. It also separates the sending of a probe from the receiving of the telemetry data, so telemetry data can be directly forwarded to management entity [26].

Although INT is not yet a widely used industry standard, it has already been applied to a number of non-constrained networking platforms; e.g. SDN, Cloud [12], [27], [28]. However, INT operations, data models, header formats, encoding and data types defined by P4 (as defined in [29]) were not designed by considering the limitations and characteristics of the wireless and/or constrained networks and the needs of Industrial WSNs and 6TiSCH Networks.

III. In–Band Network Telemetry in 6TiSCH Networks

As described in the previous section, network monitoring is a vital mechanism for 6TiSCH Networks, as any IWSN, where nodes are continuously or periodically monitored to ensure their functioning, detect relevant problems, apply performance management and network optimization. Especially, centralized route and schedule management tasks in 6TiSCH networks require collecting and analyzing measurement data, often as close to real time as possible.

This section presents the design of a light-weight, flexible and efficient in-band monitoring solution for 6TiSCH networks with minimized resource and communication overhead. The high-level overview of the resulting INT-enabled 6TiSCH Network architecture is provided in Figure 2. As illustrated in this figure, the telemetry data is collected while a packet is traversing towards the Backbone Router. When the packets reach the edge (backbone router) of the network, the telemetry metadata is removed and telemetry reports are generated to be used by any Monitoring or Management Entity for further visualization and analysis.

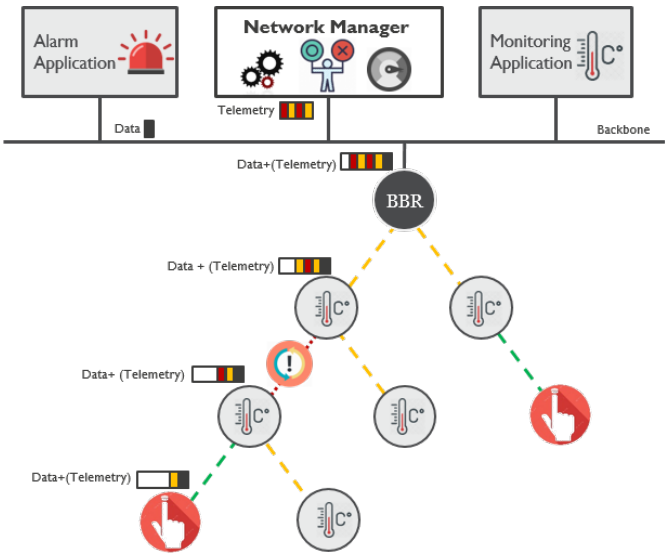


Fig. 2: INT in 6TiSCH Network Architecture.

A. Capacity-Neutral Network Monitoring

In a TSCH network, time is globally synchronized and is sliced up into time slots. The time synchronization in the network means that all nodes share a timeslot counter (encoded using 5 bytes), named Absolute Slot Number (ASN), for the total number of slots which have passed since the network has started [3]. The overall communication is orchestrated by a schedule which instructs each node what to do in each timeslot [3]. In this TSCH schedule, a single element, named *cell*, is identified by a pair of *slotOffset* and *channelOffset*, which is used to define the communication time and frequency.

The duration of a time slot is not defined by the standard, but it is defined to be long enough to send a data frame, handle the radio turnaround and receive an ACK, typically 10ms. With radios that are compliant with IEEE 802.15.4 operating in the 2.4 GHz frequency band, a maximum-length frame of 127 bytes is considered which takes around 4 ms to transmit [30]. Whatever size that a node is sending, the resources are reserved for that node so that it can transmit a data frame of 127 bytes. If the node has a shorter frame to send, there will be remaining time for that node to sleep or stay idle. That means the reserved time/bandwidth resources are wasted, instead of being used for other good reasons. Therefore, this paper proposes a mechanism that collects the monitoring information for each node by piggybacking telemetry information on the data packets in order to leverage these remaining resources, as presented in Figure 3. If there is no or insufficient remaining space in the transmitted frame, the node cannot add any telemetry information.

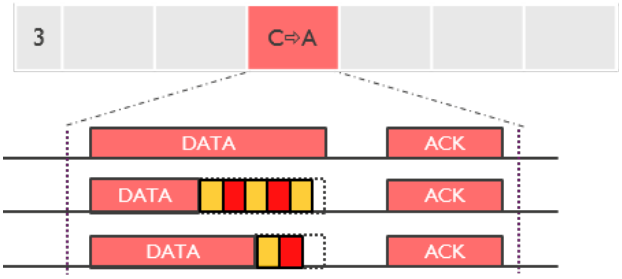


Fig. 3: Capacity-Neutral Network Monitoring

Regarding the cost of the INT operation, there will be only a limited amount of extra energy consumption for the transmitting and receiving nodes in order to transmit/receive extra bytes in the frame. However, it will not use any resource (i.e. slot, bandwidth) reserved for other application or control traffic and it will not have any effect on the network capacity, network behavior and traffic flows.

B. INT Information Elements

For the insertion of telemetry data in 802.15.4 MAC frames, the Information Elements (IEs) fields defined in IEEE Std 802.15.4 is used. The IEs are intended to extend 802.15.4 in an interoperable manner and they can be exchanged between one-hop neighbors or forwarded for communication between far devices, thus allowing several optimizations [31].

As it is presented in Figure 4, the general 802.15.4 MAC frame format defines IEs to be between the end of the MAC Header and the Frame Payload. The IEs are structured containers as Type, Length, Value fields (TLV) and they have two types, named Header IEs and Payload IEs [32]. Header IEs are the part of the MAC header and according to [32], most of their processing is done by the MAC and IETF protocols should not have any direct effect on that processing. On the other side, Payload IEs are the part of the MAC payload and they may be encrypted and authenticated. According to the standard, each frame can include one or more Header or Payload IEs that will contain information.

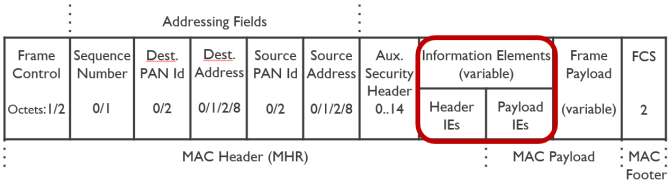


Fig. 4: General 802.15.4 MAC Frame Format.

Actually, there are already a few Payload IEs defined and used in TSCH-based technologies as sub-IEs of MAC Layer Management Entity (MLME), such as TSCH Synchronization, Timeslot and Channel Hopping IEs [33]. Also, IETF has formulated a request towards the IEEE 802.15 Assigned Numbers Authority (ANA) to allocate a registry number and described how IETF IEs should be formatted with their subtypes [32]. Also, 6TiSCH WG has expressed the need for IEs and a

temporary assignment is already provided [32]. For the design of IEs for INT data, an IETF INT sub-IE type is created by following the IETF IE subtype format.

For inserting an INT sub-IE in a MAC frame, the node first needs to set the "Information Elements Present" field in the 802.15.4 header. Next, Header IEs will be added which will be terminated by a *Header Termination 1 IE* (2 Bytes). If there is no Header IE, the *Header Termination 1 IE* will still be added in order to indicate the start of Payload IEs [34]. After that, the IETF IE descriptor (2 Bytes: type, id, length) will be added, where the IETF IE Group ID is assigned as 0x5 in IEEE 802.15 ANA [33]. Then, the INT sub-IEs can be added including the INT sub-IE descriptors (1 Byte: sub-IE ID) and the relevant INT data. At the end of the payload IEs, a *Payload Termination IE* (2 Byte) will be added. Considering all these necessary IEs, 7 Bytes of overhead will be added to the frame in order to insert any size of INT data.

The following subsections describe the approach and format for embedding telemetry information in the body of an active data packet via IETF INT sub-IEs.

C. INT Sub-IE Format

The INT-extended packets in transit need to contain telemetry instructions, so the network nodes can process and insert relevant telemetry data according to these instructions when processing the packets. In this regard, based on the requirements and targeted telemetry functionalities for 6TiSCH networks, the INT sub-IE format is designed with its headers and content, as illustrated in Figure 5. In this format, the Subtype Id represents the IETF IEs subtype identifier as defined in [32]. Currently, *202* is used as INT subtype identifier which is reserved for experimental uses.

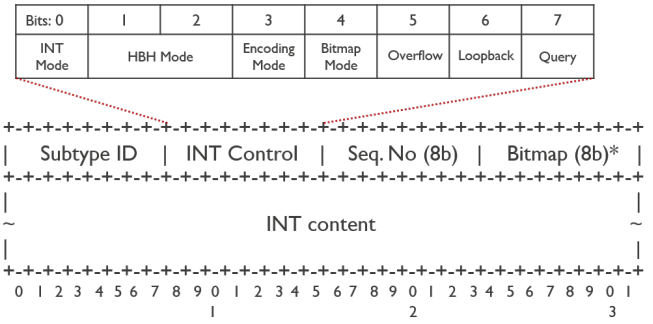


Fig. 5: The format of the IETF INT IE Subtype.

The INT Header consists of three parts; INT Control header, Sequence Number and Bitmap. The INT Control header will be used to instruct the other nodes about the telemetry modes and functions considered in the particular packet. The sequence number is an 8-bit counter for the INT source, in order to differentiate between different INT data entries from the same node and to detect the end-to-end delivery ratio for data packets with INT entries. Finally, the Bitmap is the optional INT request vector where each bit represents another type of INT data. It is used to inform middle nodes about the relevant telemetry data to add or determine the content of the INT metadata during the decoding. The details of the INT control header is provided in the remainder of this subsection.

INT Mode (1b) is used to define the mode of telemetry operation: End-to-End or Hop-by-Hop. In End-to-End mode, the middle nodes are only forwarding the INT data without any processing or addition. This mode is used to monitor end-to-end network performance or notify a central entity about local performance issues. On the other hand, Hop-by-Hop mode is used to perform per-hop telemetry operation which allows all or a subset of the traversing nodes to add telemetry data if any space is left in the current frame.

HBH Mode (2b) defines the behavior of middle nodes in Hop-by-Hop telemetry operations. It needs to be 0 if End-to-End INT Mode is selected. If Mode 1 (Opportunistic) is selected, then all the nodes will try to add telemetry data in a opportunistic manner. Mode 2 (Probabilistic) will trigger the middle nodes to follow a probabilistic approach for telemetry addition. So the nodes will add telemetry data with a certain probability which can dynamically change based on the last time it added a telemetry, the available space in the forwarded frame and the remaining number of hops. This approach can be beneficial when attempts to add INT data frequently lead to frame size overflows and can enable collecting data from a more diverse set of nodes in the network. Finally, Mode 3 enables middle nodes to decide to add or skip telemetry data in distributed manner. In this mode, the nodes which detect performance drops/issues can add telemetry data to packets as a middle node. This will also avoid the usage of resources for already known/not important data.

Encoding Mode (1b) determines the encoding mode that will be used in the INT content. The first option is using the Bitmap mode (Content or Node) which is followed by telemetry data as byte array. The type of each data will determine the length of that field which will be used to process/decode the data. The second option is using a TLV encoding, where each entry is encoded with its type, length and value. This will bring flexibility to insert data with variable length and enable nodes to decide on the INT content to insert. In order to reveal the owner of each INT entry, each node needs to add a *Node Id* entry before the other telemetry data. In addition, the whole INT content should be processed to understand what kind of telemetry data is added by each node.

Bitmap Mode (1b) defines what kind of bitmap will be used: Content Bitmap vs Node Bitmap. If it is Content Bitmap, then that bitmap will apply for each node that adding INT data. Each node will follow the given bitmap and concatenate the relevant entries to the end of the current INT content. The content needs to include all fields mentioned in the bitmap with correct sizes. So, the bitmap can be used to detect the length of each field during decoding. Alternatively, the Node Bitmap option enables each node to add its own bitmap along with the INT data which will bring independence to nodes for adding different kinds of INT data. During decoding, each node bitmap can be used to detect the length of each field.

Overflow (1b) states if any INT entry overflow has happened until that particular hop. If it is set, all of the following hops will know that they won't be able to add any INT entry, and so they can avoid any kind of INT processing.

Loopback (1b) can be used by the central entity to achieve downlink INT operation towards an end node. The central

entity can insert an INT sub-IE entry with enabled loopback and then middle nodes will add INT data until it arrives at the destination node. After that, that node will forward the collected INT data to the central management entity in any of the following uplink data messages as INT entry. This downlink INT operation will still happen fully in-band.

Query (1b) can be used by the central unit to trigger an uplink INT operation with given configuration. When a node receives a packet with attached INT sub-IE including *Query* bit set, then it will create an INT operation using the received bitmap. This can be used to create a polling-based INT operation triggered by central entity. For instance, there can be the case that the central management entity detects a problem in the network, but there is not sufficient data to troubleshoot or isolate it. Then it can send a query to certain nodes to collect more insight about the problem.

D. Telemetry Data Model

Based on a number of monitoring and management scenarios for 6TiSCH Networks, a number of Telemetry Data types are defined. The proposed telemetry data model with limited scope is provided in Table I with details about their bitmap id, name, size and description. One can extend the INT Metadata by defining any relevant telemetry data types in order to collect other network status information; such as link quality, number of neighbors, number of incoming/outgoing cells, number of re-transmissions. As it is shown in Table I, four of the bitmap ids are reserved for any further type definition.

TABLE I: Telemetry Data Model

Bitmap ID	Name	Size	Description
0	Node ID	2B	Device identifier (e.g. 802.15.4 16bit short address)
1	Receive Channel & Timestamp	2B	Channel (4b) & Reception or Generation time (12b)
2	Utilization indicator	1B	Transit Delay (4b), Queue Depth (4b)
3	RSSI	1B	Received Signal Strength (-127...0...127)
4-7	Reserved	-	Reserved for other telemetry data types

Node Id is one of the fundamental telemetry information types and represents the unique identifier of the node that inserts the telemetry data. In the scope of 6TiSCH networks, IEEE 802.15.4 16-bit short addresses can be used.

Receive Channel and **Timestamp** constitute a combined telemetry entry. The first 4 bits of this field represent the channel (0...15) the packet is received on, i.e. one of the available 16 IEEE 802.15.4 channels. The Timestamp represents the 12 least significant bits of the time (expressed in ASN which is 5 bytes) at which a packet that needs to be forwarded is received. For the source node, this time represents the time the packet is generated. Since all of the network nodes share the same ASN, the timestamps on each node are inherently synchronized. Assuming a 10ms slot length, 12 bits are enough to represent 40.96 seconds which is sufficient to detect all of the timestamps based on the reception ASN at the border router. The packet generation time can also allow us to understand the age of telemetry data and evaluate its validity.

Utilization indicator illustrates the node occupation when the packet traverses that node. The first 4 bits of this field represent the transit delay which is the delay (in slots) between the reception of a frame and its entry to the outgoing queue to be transmitted to the next hop. For the source node, this field will be 0. The remaining 4 bits constitute the Queue Depth value which is the number of packets in the outgoing queue at the time.

RSSI represents the received signal strength for that frame measured at the particular hop. It can take values between -127 dBm and 127 dBm. This value is 0 for the source node and will be ignored during INT processing.

E. Telemetry Operations

Thanks to the flexible and powerful design, the proposed INT solution offers several methods and operations for various network elements to collect and report their state in near real-time, allowing for maximized visibility and improved cooperation in 6TiSCH networks.

Firstly, it offers hop-by-hop telemetry which can collect per-hop latencies, queue states and link qualities and can be used to detect which rules a packet followed and to detect how long a packet was queued at each node. On the other side, it is also possible to obtain real-time edge-to-edge packet-level network information (e.g. reliability, latency) based on INT sequence numbers and timestamps at source and sink node.

Secondly, the proposed solution provides real-time monitoring capabilities where the collected telemetry data reflects the momentary network performance and the exact treatment that an application packet encounters. In contrast, the probing and polling monitoring data would probably represent a historical average or data representing the situation at probing time, not at the data transmission time. Also, probing data from different nodes would arrive at different moments and so could not reflect simultaneous network snapshots, which may prevent the isolation of network problems.

The proposed INT solution also offers flexibility in terms of telemetry initiation and addition approaches: continuous, periodic, event-driven or query-driven. So source nodes or middle nodes can perform continuous monitoring by adding telemetry data to every data packet if the frame size allows. Alternatively, they can add telemetry data only with a certain periodicity. Moreover, it also enables network nodes to add telemetry and inform management entities upon a network event, such as local performance drops, congestion or a new QoS request. If the unexpected incident needs to be notified to the central unit immediately, the node can also choose to generate an empty data message which results in a probing-like telemetry. Finally, via the introduced *Query* option in the INT header, Query-driven telemetry can be achieved upon a request from a central monitoring entity.

Unlike existing in–band monitoring technologies, the proposed INT design offers both uplink and downlink telemetry operations which enables monitoring applications to monitor the network performance in both directions.

Another offered feature is that forwarding middle nodes can initiate an INT operation on a packet with another source. So,

each network node which generates or forwards data packets can become a source of INT data. Moreover, by means of the flexibility to decide what to add, the nodes are enabled to add a subset of INT entries if not all of them fit in the frame. These two features allow distributed and intelligent INT Strategies.

Finally, regarding the security of the INT entries, the INT protocol does not define its own security mechanisms. However, since INT fields are carried as Payload IEs, they can be encrypted and authenticated through link-layer security through *CCM** [35] with the same level of security as any other Payload IE.

IV. Design and Implementation

In order to validate and demonstrate the proposed telemetry and monitoring solution, we implemented its fundamental functionalities in a widely used operating system for embedded IoT and wireless sensor devices with IPv6 connectivity: *Contiki–NG* [36]. Contiki–NG is an open-source, cross-platform operating system for IoT devices and it focuses on dependable IoT and low-power communication protocols such as 6LoWPAN, RPL, and CoAP [36]. Recently, it has also been extended with support for TSCH and 6TiSCH with so-called "minimal configuration" [37] along with a simple Scheduling Function (SF) (*sf-simple*) which provides APIs for user processes in order to add or remove cells dynamically.

This section briefly describes the implementation details of the INT solution and discusses how it is integrated into the 6TiSCH Stack along with a Network Monitoring Application.

A. INT Sub-Layer

In order to achieve INT, the 6TiSCH protocol implementation is extended so that nodes can add, process and forward telemetry data attached to 802.15.4e frames. Figure 6 illustrates the 6TiSCH Network Stack complemented with INT facilities. It also provides details about INT sublayer which is responsible for telemetry operations at each node, as part of packet generation or forwarding.

First of all, this sublayer cooperates with higher and lower layers in order to retrieve telemetry data (timestamps, RSSI, etc.) and relevant attributes (destination, packet type, etc.) about the processed packet. During the packet reception process, the lower layers need to save the received INT sub-IEs along with measured telemetry data (i.e. timestamp, reception channel, RSSI) and deliver these metadata to INT sublayer.

Upon the arrival of an outgoing packet in the INT sublayer, it will first check its suitability for INT insertion. The INT information won't be added to frames which carry 6LoWPAN fragmented messages, broadcast messages, RPL and 6TiSCH control messages. Broadcast messages and control messages probably won't be populated towards the border–router. For the fragmented messages, each packet is fragmented and defragmented at each hop along the path. Most of the fragments, except for the last one, will be already full. A strategy could be to have smaller fragments with INT data and keep the INT data along the fragments. But, since this would impact the network behavior, it is not considered in the resulting design.

Next, if the packet is a packet that needs to be forwarded, the INT sublayer needs to check if there is an INT Sub-IE available in the packet. If there is, the INT Header will be processed by the *INT Header Handler*. This handler will check the INT mode and the overflow field. If the requested INT operation is End-to-End or if there was already an overflow in the previous hops, the INT sublayer will forward the packet without any changes on the INT entries.

If the packet incurs Hop-by-Hop mode, then the resulting frame size will be estimated by the *Frame Size Estimator* in order to validate that INT insertion will not lead to an overflow. If inserting a new telemetry record causes *Current Length* to exceed *Max Length*, then no record should be added and the node should *Skip* the INT addition and set the overflow flag. If not, it will notify the INT Engine to perform INT operation with given Bitmap and remaining frame size available for INT entries. Due to IP Header Compression in 6LoWPAN Networks, Source and Destination Address can be elided (link-local) when either matches the compressible IP prefix, so the size of the frame might change while traversing the network [38]. For certain packet sizes, this may also lead to an overflow in the middle nodes for inline INT entries. In such cases, middle nodes need to remove the INT content or the entry completely from the forwarded messages. If the INT header fits, then the INT sublayer can keep the header and remove only the INT content. Also due to 6LoWPAN packet size changes, the frame might get fragmented by the middle nodes which will again lead to the removal of the INT information.

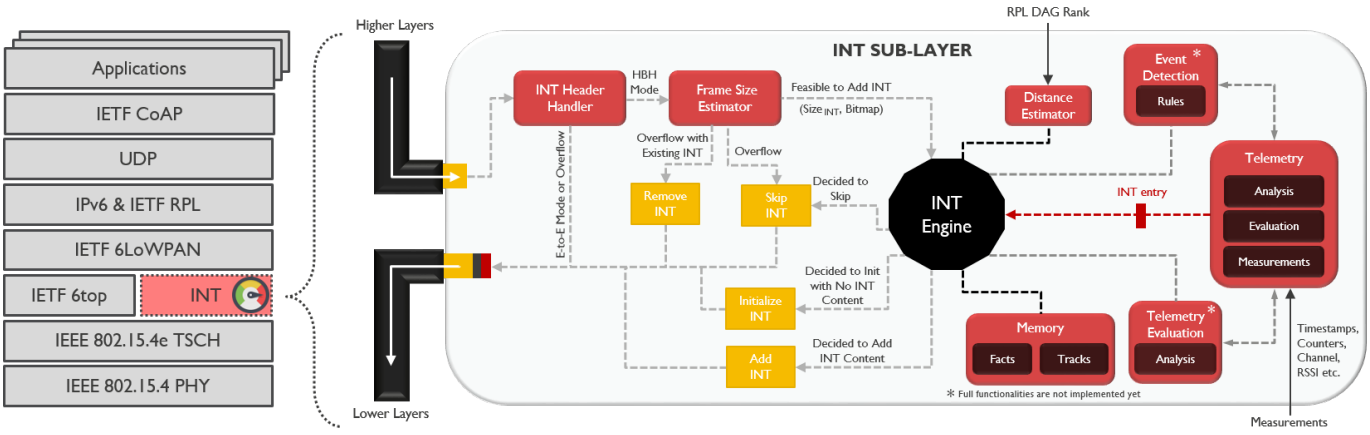


Fig. 6: 6TiSCH Network Stack with INT Sub-Layer.

The *INT Engine* is the core processing entity which evaluates and decides the insertion of INT entries based on several factors including INT mode, encoding mode, INT Bitmap and remaining space in the frame, etc. Based on the decision of the INT Engine, the node will *Skip* the INT or add INT header to *Initialize* INT operation or *Add* INT content along with the header. During this process, the INT Engine can make use of several locally calculated indicators and follow several INT insertion strategies as further discussed in Section V-C.

Firstly, the remaining hop count can be taken into account to achieve smarter INT operations. This will be estimated by the *Distance Estimator* using the RPL DAG Rank and RPL link parameters (i.e. MaxRankIncrease, MinRankIncrease, RootRank). Secondly, the INT Engine can make use of the INT history and facts collected at the *Memory* and may avoid adding repetitive INT entries by checking the last time a similar INT operation is performed. Besides, two other entities are envisioned (full functionalities are not implemented) in INT Sub-Layer: the *Event Detection* and *Telemetry Evaluation* entities. The former one can monitor the local performance metrics and detect events or misbehavior and notify the INT Engine in order to trigger an INT operation. While, the latter entity can continuously process all telemetry data collected by the *Telemetry* entity and assign an importance/relevance metric to each of them which can be used by the INT Engine during the INT addition and content selection process.

B. Network Monitoring Application

Based on the INT extensions to Contiki-NG, a Network Monitoring and Analysis Application is created which can collect and analyze INT data and extract insights about the network performance in order to monitor network topology, hop-by-hop utilization and latency performance, end-to-end latency and reliability and also communication schedules at each node.

A sample screenshot of the resulting dashboard from the Monitoring Application is provided in Figure 7. The first subfigure in this dashboard illustrates the discovered network topology, where the color of each node represents the utilization of that node based on the most recent INT metrics. Also, the trace of the latest INT message is shown with wider lines with different colors which indicates the measured RSSI on the given link. The second subfigure presents the discovered communication schedules on each node based on the timestamps, used channels and the channel hopping sequence used by the network. After that, the third subfigure shows the real-time end-to-end delay measurements for each node over time. Finally, the last subfigure shows the number of INT entries collected from each node with a separate justification of generated and forwarded packets. As it is shown in this figure, the middle nodes and the sink nodes provide relatively more telemetry data due to the forwarded messages.

V. Evaluation & Validation

This section firstly presents the evaluation of the proposed telemetry solution in terms of energy efficiency and how it outperforms the traditional probing or polling–based alternatives. Next, it demonstrates that the application traffic and network behavior is not affected by the INT operation, unlike probing approaches. Then, it discusses about the opportunistic and probabilistic INT strategies and presents their performance evaluation in a basic multi-hop setup. Finally, the results of testbed experiments are presented in order to validate and evaluate the INT mechanism in real-world networks.

A. Energy Efficiency

In order to demonstrate the efficiency of the INT operations, a theoretical study is performed regarding the energy consumption resulting from the telemetry operations. Then the INT mechanism is compared with the probing and polling-based approaches in terms of total energy consumption.

First, the cost of the transfer of INT data (including INT header and content) is calculated, with varying telemetry sizes, over a single hop. For this, this study considers the extra consumption in order to send and receive more data in a single 802.15.4 frame by the transmitting and receiving node, respectively. It is assumed that both the transmitting and receiving node could go to sleep mode (CPU and radio sleep) for the duration of the extra INT transmission in case they were not transmitting any INT data. That duration constitutes the extra energy consumption and the resulting extra charge (in coulombs) drawn from the battery due to INT transmission over a single hop, Q_{INT}, and is represented by:

$$Q_{\text{INT}} = \Delta t_{tx}.(I_{tx} - I_{sleep}) + \Delta t_{rx}.(I_{rx} - I_{sleep}) \qquad (1)$$

where Δt_{tx} and Δt_{rx} are the extra transmission and reception duration due to INT insertion. These values are equal and

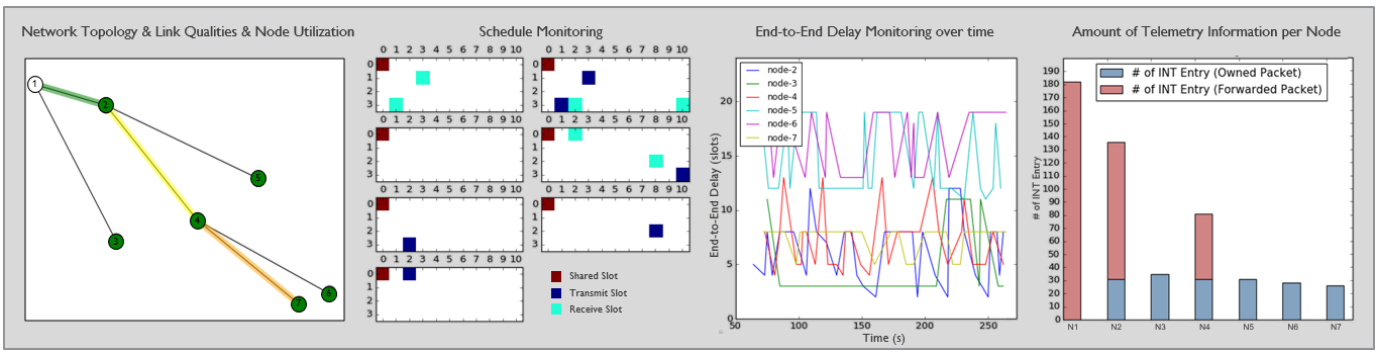


Fig. 7: Network Monitoring Dashboard.

they can be calculated by dividing the INT data size by the data transmission rate: S_{INT}/R_{tx}. In this equation, I_{tx}, I_{rx} and I_{sleep} are the current drawn by the node in transmission (CPU sleep, radio in TX mode @0dBm), reception (CPU sleep, radio in RX mode @-50dBm input power) and sleep state (CPU and radio sleep), respectively. For these calculations, the current (I) values for the CC2538 System-On-Chip for 2.4–GHz IEEE 802.15.4 [39] are used based on the measurements in [40]; I_{rx}: 23.16mA, I_{tx}: 27.55mA, I_{sleep}: 10.06mA. The calculated consumption values for the transmitting node, receiving node and the total is presented in Figure 8a. This figure shows that the energy consumption is linearly increasing with the size of the INT data.

Based on these calculations, it is possible to investigate and compare the total amount of charge that is spent to deliver telemetry data with varying sizes to the sink, considering three different approaches: INT, Probing and Polling. For INT, the total extra charge drawn from the network nodes to deliver an INT entry with a certain length from a node with particular depth/distance (d) from the border router can be calculated as follows:

$$Q_{\mathrm{INT}_{total}} = \sum_{hop=1}^{\mathrm{d}} Q_{\mathrm{INT}} = \mathrm{d} \times Q_{\mathrm{INT}} \tag{2}$$

For the Probing approach ($Q_{\mathrm{pr}_{total}}$), it is assumed that the transfer of each probe (Q_{pr}) will result in a new 802.15.4 frame and an acknowledgment transmission on each hop. For the Polling approach ($Q_{\mathrm{pl}_{total}}$), a poll request (Q_{req}) is considered which needs to be transferred from the central entity to the end-node and which results in an uplink response message (Q_{resp}) including telemetry. The formulas for calculating the consumed charges in probing and polling approaches are as follows:

$$Q_{\mathrm{pr}_{total}} = \sum_{hop=1}^{\mathrm{d}} Q_{\mathrm{pr}} = \mathrm{d} \times \left[Q_{\mathrm{tx}}(S_{\mathrm{pr}}) - Q_{\mathrm{sl}} + Q_{\mathrm{rx}}(S_{\mathrm{pr}}) - Q_{\mathrm{idle}}\right] \tag{3}$$

$$\begin{aligned} Q_{\mathrm{pl}_{total}} &= \sum_{hop=1}^{\mathrm{d}} Q_{\mathrm{req}} + \sum_{hop=1}^{\mathrm{d}} Q_{\mathrm{resp}} \\ &= \mathrm{d} \times \left[Q_{\mathrm{tx}}(S_{\mathrm{req}}) + Q_{\mathrm{rx}}(S_{\mathrm{req}}) + Q_{\mathrm{tx}}(S_{\mathrm{resp}}) \right. \\ &\qquad \left. + Q_{\mathrm{rx}}(S_{\mathrm{resp}}) - 2 * Q_{\mathrm{idle}} - 2 * Q_{\mathrm{sl}}\right] \end{aligned} \tag{4}$$

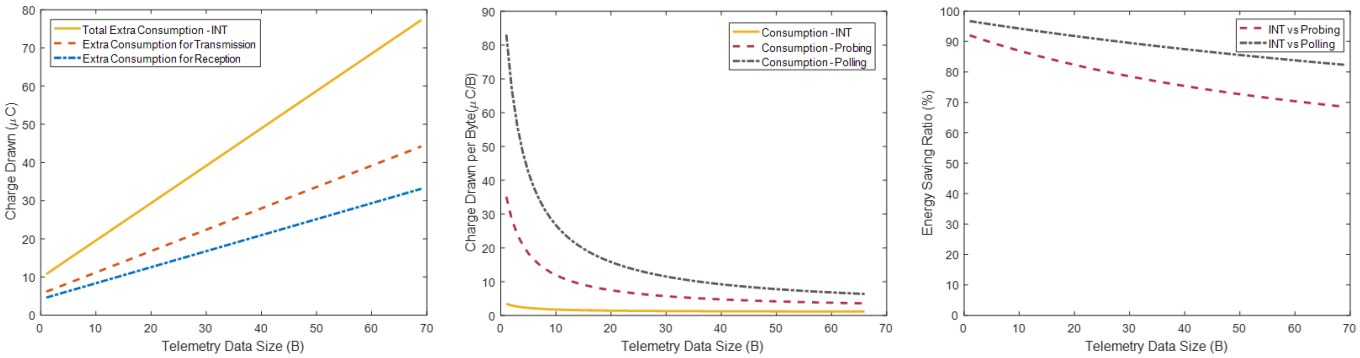
where S_{pr}, S_{req} and S_{resp} are the payload size for probe, polling request and response, respectively. While Q_{tx}, Q_{rx}, Q_{idle} and Q_{sleep} represents total charge drawn in transmitting (TxDataRxACKbase), receiving (RxDataTxACKbase), receiving idle (RxIdle) and Sleep slot operation for certain frame sizes. The values are based on [40].

The average charge consumption per telemetry byte per hop for varying telemetry data sizes in 3 approaches are presented in Figure 8b which is complemented by Figure 8c which illustrates how much energy one can save by using INT instead of the probing and polling approaches. These results show that the INT approach enables us to transfer telemetry data with minimal network cost and achieve remarkable energy-saving between 70-90% compared to probing and polling approaches. The main reason for the energy-saving here is that probing and polling mechanisms generate control packets which include all packet headers and require hop-by-hop acknowledgments.

After all, by considering the total cost of an INT operation to the network and the importance/relevance of the telemetry data, the node can decide to add or skip the INT entry.

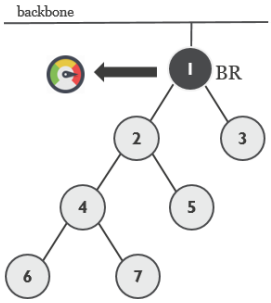
B. Network Traffic Impact

In order to investigate the effect of the INT operation on the network performance, a series of tests are performed by using Cooja, a simulation software for wireless sensor network applications with Contiki motes. We first run an application network without any telemetry or monitoring messages, where each node generates an application packet (payload: 1-32 bytes) with random delays [0.1, 1.1 seconds]. This results in an average application data generation frequency which the network can handle. Then, the same network is simulated, but with the insertion of INT and monitoring messages, with continuously increasing frequencies, generated by each node. For the INT operations, it is assumed that all of the defined telemetry metrics are monitored, resulting in 6B data insertion in each hop. For monitoring application, it is assumed that 10B monitoring data is sent as part of probes by each node towards the border router. For the TSCH scheduling, the *sf-simple* scheduling function is used which schedules each node with a single outgoing cell towards the parent and maximum of six cells in total including the incoming cells from each RPL child. The simulation setup and detailed network parameters and application settings are provided in Figure 9.



(a) Charge Drawn for INT. (b) Charge Drawn per Byte per Hop. (c) Energy Saving Ratio with INT.

Fig. 8: Energy efficiency calculations for INT, Probing and Polling-based Telemetry.



Parameter	Value
Number of Sensor Motes	6 + 1 BBR
Data Generation	0.1 - 1.1 second
Queue Size	16
Time Slot Duration	10ms
Length of Slotframe	11
Number of Shared Slots	1
Monitoring Message Size	10B
Max Payload	32B
802.15.4 Channels	[15,20,25,26]
Duration	20 minutes

Fig. 9: Simulation setup with detailed settings.

Using the given setup, we performed measurements on the total application throughput for all scenarios, along with the amount of monitoring and INT data collected. The results of these measurements are presented in Figure 10.

As it can be seen in Figure 10a, when there is no inserted monitoring or telemetry data, the network can provide a stable application throughput around 10kB/min. However, as Figure 10b shows, the application throughput decays when monitoring probes are generated by network nodes, while the total network throughput stays similar around 10kB/min. This is due to the fact that the network bandwidth and communication resources/slots are shared by both application and monitoring traffic. When the frequency of monitoring data becomes very high, it starts to dominate the network resources which may even prevent connectivity for the application traffic.

Besides, as illustrated in Figure 10c, INT operations do not impact the application data traffic and the network is able to achieve similar application throughput with INT, even with higher telemetry insertion rates. In contrast, the total network throughput has increased up to 18 kB/min with high INT insertion rates. This shows that the proposed INT method is leveraging the resources which were already assigned, but were not going to be used to the fullest. After that point, although the frequency of INT reporting is increased, the amount of collected INT is not increasing, as the available network resources are fully utilized.

These results show that unlike the probing approach, INT can provide network telemetry information without any effect on the behavior and performance of the network.

C. Evaluation of INT Strategies

In this section, two of the possible INT insertion strategies are studied: Opportunistic and Probabilistic. Firstly, these strategies are described along with a discussion regarding their suitability in different network scenarios, then their performance and behavior in a simple 6TiSCH Network is presented.

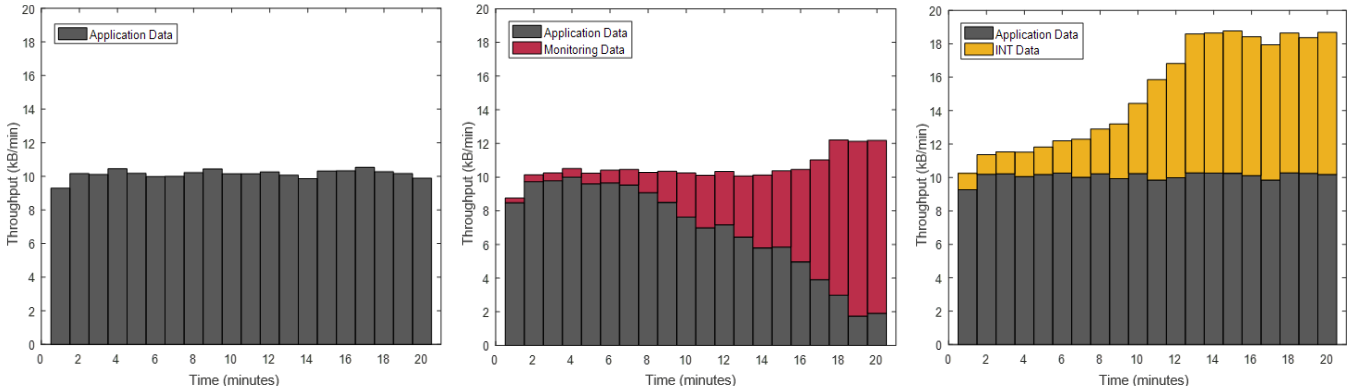
1) Opportunistic Logic: In this strategy, each node tries to exploit immediate telemetry insertion opportunities, regardless of any planning or principle, in a greedy manner. So, the nodes will take every chance to insert telemetry in any suitable outgoing packet towards the border router. Although this approach will maximize the total amount of collected telemetry, the source node and the nodes which are closer to the source will have a higher chance to insert telemetry data and subsequent nodes may not even get any chance to add any telemetry. Therefore, for certain network scenarios, especially for large networks with limited telemetry opportunities, this approach may result in an inadequate network view due to the telemetry information that comes from only a limited part of the network.

2) Probabilistic Logic: In this strategy, the nodes are following a probabilistic approach where each node inserts or skips INT entries with certain probabilities which are dynamically calculated in a distributed manner. Although this approach may result in a lower amount of telemetry data, it results in a better distribution of the telemetry data across nodes and thus a more diverse set of telemetries and a more clear/wider network image.

Regarding the INT insertion probabilities p_i, the following formula can be used:

$$p_i = 100 \times \underbrace{\left\lfloor \frac{MTU - S_f}{S_{int+}} \right\rfloor}_{\text{\# of possible INT entries}} \Big/ \underbrace{\left\lfloor \frac{R}{\Delta r_{min}} \right\rfloor}_{\text{Remaining hop count}} \tag{5}$$

which dynamically calculates the telemetry insertion probability based on the current frame size S_f (including headers, payload, current INT), the size S_{int+} of a newly to be added INT entry based on Bitmap, and the current RPL Dag Rank R. In this formula, MTU and $\Delta\ r_{min}$ represent the Maximum Transmission Unit and RPL Minimum Rank Increase configuration, respectively.



(a) Application Throughput. (b) Application and Monitoring Throughput. (c) Application and INT Throughput.

Fig. 10: Network Throughput Measurements in different scenarios.

Figure 11 provides an overview of the opportunistic and probabilistic INT strategies along with their performance (INT collection distribution and inter-arrival time) in a simple network setup simulated in Cooja. In the given setup, node 4 is the source node which creates application traffic towards an application server on the backbone network. Node 2 and 3 are relay nodes and they do not generate any traffic. The border router, node 1, is at the edge of the 6TiSCH Network where the telemetry data is collected. Therefore, the border router can always add telemetry information (e.g. reception time and channel) before it forwards it to the monitoring application via a non-constrained network.

As presented in Figure 11a and 11b, when using the Opportunistic INT Strategy, the source node will always have more opportunities to insert telemetry data compared to relaying nodes. Especially, node 2 will only have a chance to insert telemetry in case there is any remaining space left after the insertion of INT data by the source and hop 1. This results in an unfair telemetry distribution and different INT inter-arrival times for different nodes. As it can be seen in Figure 11b and 11c, the Source node had almost 5 times more telemetry data compared to Hop2 and the average INT inter-arrival time performance for Source, Hop1 and Hop2 were significantly different with 668 ms, 1112 ms, and 3086 ms, respectively.

On the other side, using the probabilistic approach, each node has an equal opportunity to insert telemetry data, despite their different distances. This approach was expected to result in a more fair and distributed number of INT data collected at the network management entity. As presented in Figure 11d, 11e and 11f, the outcome of the measurements also reflects this, where Source, Hop1 and Hop2 had similar insertion ratios (around 50%) and average INT inter-arrival times: 1131 ms, 1151 ms and 1240 ms respectively.

D. Testbed Evaluation

In this subsection, the results of basic testbed experiments are presented that was conducted in order to validate and evaluate the INT mechanism in real IWSNs. For this purpose, several tests are performed in various scenarios in two wireless testbeds, named the Wireless Testlab (w-iLab.2) and the OfficeLab (w-iLab.1), which offer several industry or office-like wireless settings [41]. The utilized testbed consists of Zolertia Remote nodes that are static and do not move. A sample experimental setup and typical network topologies are presented in Figure 12. In these experiments, the INT-enabled 6TiSCH stack implementation in Contiki NG is used, along with available basic RPL multihop routing protocol.

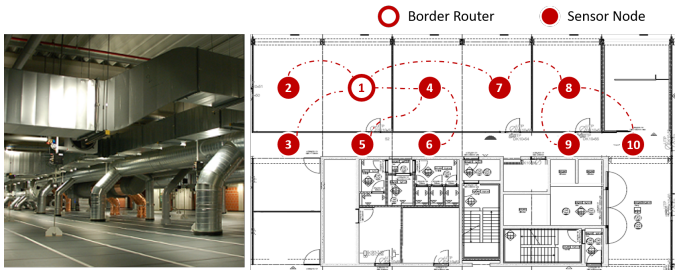
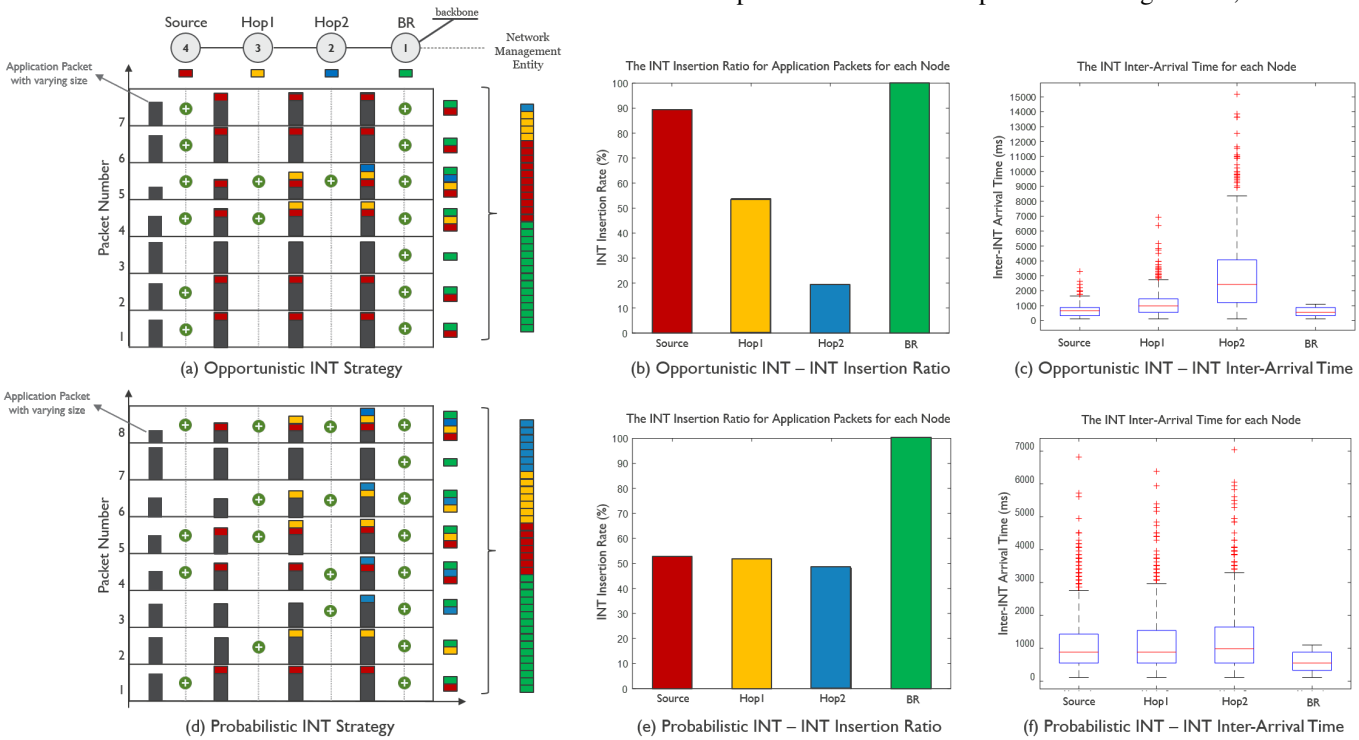


Fig. 12: Testbed Experimental Setup & Network Topologies.

The telemetry information is gathered at the border router (BR) and delivered to a monitoring application where the data is stored and processed. The data that is analyzed here was collected over several days which resulted in hundreds of thousands of INT entries, but without any extra monitoring packet exchange. Then, this telemetry data is mainly analyzed to understand end-to-end QoS, wireless link qualities, node and link utilization, and discover the network topology. Some sample measurements are presented in Figures 13, 14 and 15.



(a) Opportunistic INT Strategy
(b) Opportunistic INT – INT Insertion Ratio
(c) Opportunistic INT – INT Inter-Arrival Time
(d) Probabilistic INT Strategy
(e) Probabilistic INT – INT Insertion Ratio
(f) Probabilistic INT – INT Inter-Arrival Time

Fig. 11: The Overview and Evaluation of INT Insertion Strategies.

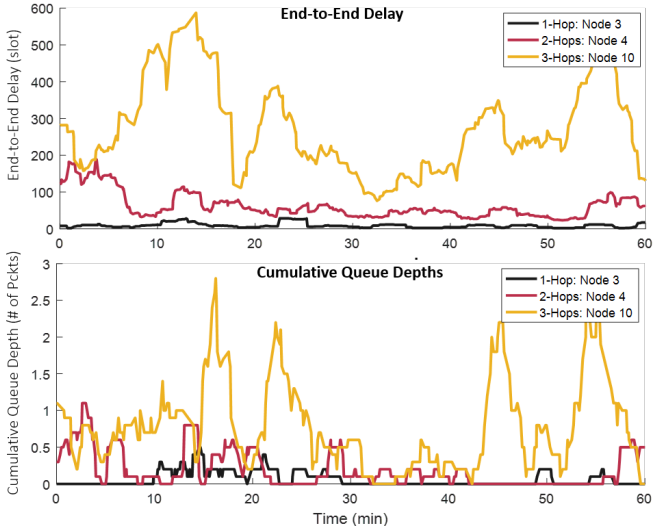


Fig. 13: End-to-End Delay and Cumulative Queue Depths.

Figure 13 presents the end-to-end delay performances and the cumulative queue depths (the sum of the reported queue depths) for several packets from three nodes with varying distances to the BR. This figure shows that as the distance from the BR increases, the resulting QoS is becoming more unstable which should be addressed by the network manager.

Figure 14 presents the RSSI reportings for Node 5 over a certain time. As it is illustrated in this figure, by using this information, one can extract route changes in the network and monitor link qualities between a certain node and its parents.

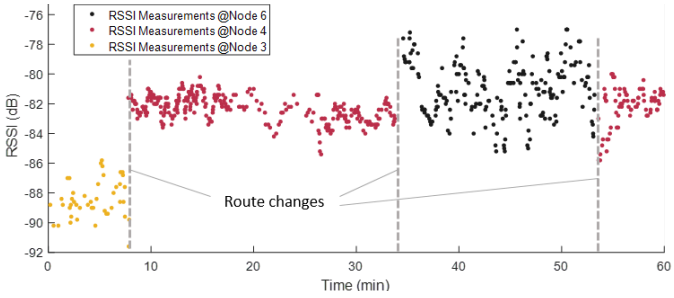


Fig. 14: RSSI Reportings for Packets Transmitted by Node 5.

Finally, the distribution of channel appearances in INT measurements is provided in Figure 15. This figure shows that although all channels have the same probability of usage, the successfully received packets (out of 100 packets) observed on channel 25 and 26 are higher than the other two channels, which reflects lower interference in these channels.

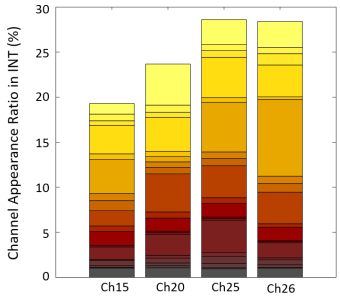


Fig. 15: Channel Appearance Distribution.

VI. INT-based Network Monitoring for IWSNs

Through its flexible and powerful design, the proposed 6TiSCH INT solution offers several methods and operations for various network elements to collect and report their state in real-time, allowing for maximized visibility and improved cooperation. For instance, based on collected *Node-Ids*, it is possible to detect which path a packet takes and perform *Network Topology Monitoring*. Based on edge-to-edge latency and reliability measurements, *QoS Validation and Service Verification* can be achieved. The per-hop telemetry data (queue, transfer delay, etc.) can be used for *Debugging & Troubleshooting* or *Link Utilization Monitoring*.

In addition to classical network monitoring operations, the proposed INT solution can also create novel network monitoring and analysis functionalities for Industrial WSNs. For instance, a *Schedule and Route Monitoring* can be performed based on Node-Ids, Receive Channels and Timestamps collected from each hop. Or, based on the monitoring of complete network view, *Network Optimization* or *Intelligent QoS-Aware Routing/Scheduling* can be realized. Or, a Network Manager can try to detect internal or external changes in the network based on limited visibility collected via INT. Or, it can try to learn/discover network configurations and settings (e.g. different TSCH configs) based on the collected information. Or, local nodes can make use of monitoring data to gain network awareness, learn network limits and adjust their operations accordingly. The remaining of this section demonstrates how the INT mechanism can be used to achieve some of these functionalities.

1) Network Awareness Enhancement and Edge-to-Edge Congestion Control: In order to harness the full potential of the network resources, wireless sensor nodes can make use of INT to explore the network limits and available resources and adjust their operation accordingly. In order to demonstrate this functionality, an Edge-to-Edge Congestion Control mechanism is created based on the monitored *Queue Depth* metrics. The created network setup and the operation of the congestion control mechanism are illustrated in Figure 16. In this setup, all nodes are generating application traffic with varying size and period and a central entity is sending notifications, based on the monitored queue states, for the nodes to adapt their generation rate to avoid congestion and buffer overflow. In Figure 16, packet generation and application throughput measurements are provided where Node 7 is first gradually increasing its packet generation rate and then adapting to avoid congestion.

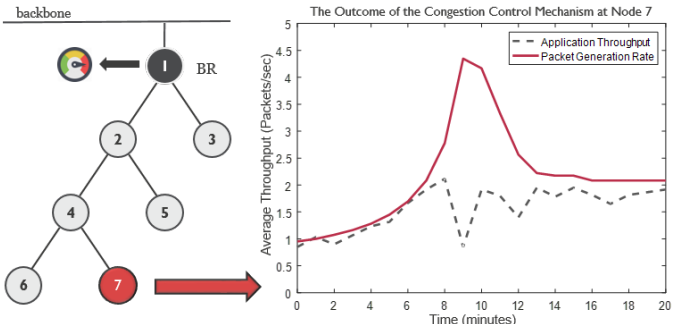
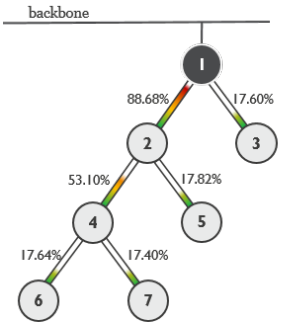


Fig. 16: Congestion control via INT.

2) Network Usage and Resource Monitoring: 6TiSCH is using a reservation-based MAC design which eases the calculation and detection of the reserved and utilized network resources at each node and link. A momentary measurement from the simulated network for link utilization and remaining resources at each node are illustrated in Figure 17. Based on this collected complete network view, the central entity can derive how close the network is to reaching network limits or bottleneck anywhere in the network. These measurements can also be used to estimate the lifetime of each node in the network.



Node Id	Resources Reserved	Utilization of Reserved Resources
1	27.27 %	53.14 %
2	36.36 %	53.20 %
3	18.18 %	17.60 %
4	36.36 %	29.38 %
5	18.18 %	17.82 %
6	18.18 %	17.64 %
7	18.18 %	17.40 %

Fig. 17: Network Utilization and Resource Monitoring.

3) Network Monitoring based on Limited Visibility: The INT operations are ultra-efficient and flexible, however, their frequencies are directly dependent on the application data traffic. Therefore, it inherently may not be possible to have full network visibility in many scenarios. However, the management entity can try to make the maximum of available limited visibility in order to predict network activities and changes in other parts of the network. If a problem or change is detected, then more powerful tools can be activated to collect more insight. In this sense, the setup in Figure 18 is created, where the central entity is only able to collect telemetry data on the path from Node 7. By just monitoring edge-to-edge latency values, the central entity can detect the impact of a network change: in this case an increase in the application traffic rate for Node 5.

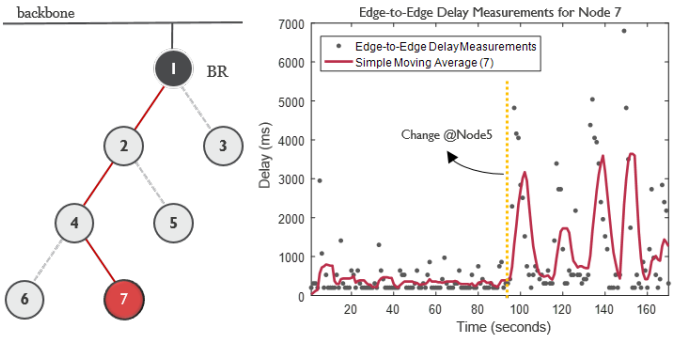


Fig. 18: Network Activity Detection in Limited Visibility.

4) Network Problem Troubleshooting and Isolation: The INT operation can also allow central entities to detect exact location and reason of a problem by combining and correlating measurements from different INT flows. In this sense, a setup with a problematic link is created as illustrated in Figure 19. The central entity can localize this problem by correlating the hop-by-hop latency measurements on the flows from Node 6 and 7. As illustrated in the measurements, Node 6 packets are facing variable and large delays in the first hop, while Node 7 packets do not have any issue. That means the problem is not at Node 4 or after, instead there is an issue with Node 6 or the link between Node 6 and Node 4.

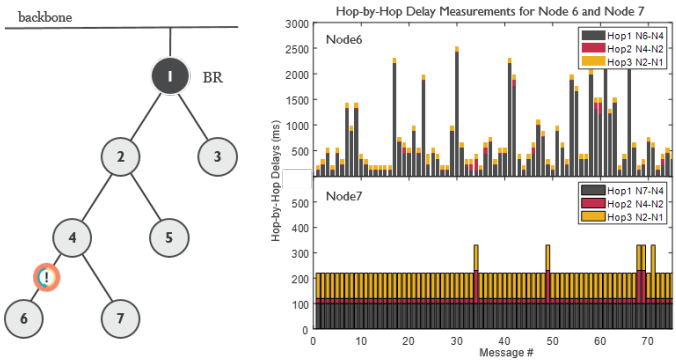


Fig. 19: Detection of Problematic Link via INT.

VII. Conclusion

In this paper, the problem of efficient network monitoring in Industrial Wireless Sensor Networks is addressed. A novel and efficient network telemetry solution is investigated, which creates a flexible and powerful in-band network telemetry design with minimized resource consumption and communication overhead while offering extensive and useful monitoring capabilities and more intelligent and complex telemetry strategies.

Particularly, a conceptual design for a novel capacity-neutral network monitoring mechanism for open standards-based 6TiSCH Architecture is presented, which was inspired by the recent In-Band Network Telemetry concept. By means of modeling, implementation, and simulations, it is demonstrated that the proposed INT approach can provide ultra-efficient network monitoring operations without any effect on the network behavior and performance, validating its suitability for Industrial Wireless Sensor Networks. Although this paper mainly describes and implements the proposed INT solution for the 6TiSCH protocol stack, the same approach and design can be also applied to any 802.15.4e-like and TSCH-based networking technologies; such as WirelessHart and ISA100.11a.

In addition, this work also showed that the proposed INT solution can enable performing a wide range of monitoring and management operations in various network scenarios, including anomaly detection, service verification, congestion control, problem troubleshooting and isolation, traffic engineering and network optimization. This way, this work can serve as a starting point for further research on intelligent and powerful network management systems for more flexible and reconfigurable industrial networking solutions targeting increasingly diverse and demanding network dynamics and requirements with a wider and more fine-grained scale.

REFERENCES

[1] IEC-62591, "Industrial communication networks - Wireless communication network and communication profiles-WirelessHART."

[2] IEC-62734, "Industrial networks - Wireless communication network and communication profiles - ISA 100.11a."

[3] "IEEE standard for local and metropolitan area networks–part. 15.4: low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer," IEEE, Standard 802.15.4e-2012, Apr. 2012.

[4] P. Thubert, Ed., "An architecture for IPv6 over the TSCH mode of IEEE 802.15.4," IETF, Internet-Draft draft-ietf-6tisch-architecture-24, July 2019.

[5] E. Grossman, Ed., "Deterministic networking use cases," IETF, RFC 8578, May 2019.

[6] T. Sauter, "Accessing factory floor data," in *Industrial Communication Technology Handbook*, R. Zurawski, Ed., June 2014, vol. 8, no. 2.

[7] N. Ramanathan *et al.*, "Sympathy for the sensor network debugger," in *Proc. of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys '05)*, 2005.

[8] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1132–1144, Aug 2010.

[9] M. Keller, J. Beutel, and L. Thiele, "The problem bit," in *Proc. of IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 105–114.

[10] K. Romer and Junyan Ma, "PDA: Passive distributed assertions for sensor networks," in *Proc. of International Conference on Information Processing in Sensor Networks*, April 2009, pp. 337–348.

[11] The P4.org Applications Working Group, "In-band network telemetry (INT) dataplane specification," The P4 Language Consortium, Tech. Rep., August 2018.

[12] A. Gulenko, M. Wallschläger, and O. Kao, "A practical implementation of in-band network telemetry in Open vSwitch," in *Proc. of 7th IEEE International Conference on Cloud Networking (CloudNet)*, Oct 2018.

[13] A. Karaagac, I. Moerman, and J. Hoebeke, "Hybrid schedule management in 6TiSCH networks: The coexistence of determinism and flexibility," *IEEE Access*, vol. 6, pp. 33 941–33 952, 2018.

[14] D. Raposo *et al.*, "Industrial IoT monitoring: Technologies and architecture proposal," *Sensors*, vol. 18, no. 10, p. 3568, Oct 2018.

[15] P. Lapukhov and R. Chang, "Data-plane probe for in-band telemetry collection," IETF, Internet-Draft draft-lapukhov-dataplane-probe-01, 2016.

[16] S. Shirali-Shahreza and Y. Ganjali, "Traffic statistics collection with FleXam," in *ACM SIGCOMM*, 2014, pp. 117–118.

[17] J. Geng, J. Yan, Y. Ren, and Y. Zhang, "Design and implementation of network monitoring and scheduling architecture based on P4," in *Proc. of 2nd ACM International Conference on Computer Science and Application Engineering*, 2018, pp. 182:1–182:6.

[18] A. Rodrigues, T. Camilo, J. Sá Silva, and F. Boavida, "Diagnostic tools for wireless sensor networks: A comparative survey," *Journal of Network and Systems Management*, vol. 21, 09 2013.

[19] S. Rost and H. Balakrishnan, "Memento: A health monitoring system for wireless sensor networks," in *Proc. of IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2006.

[20] S. Tennina *et al.*, "Z-Monitor: A protocol analyzer for IEEE 802.15.4-based low-power wireless networks," *Computer Networks*, vol. 95, 12 2015.

[21] M. Keller, J. Beutel, and L. Thiele, "How was your journey?: Uncovering routing dynamics in deployed sensor networks with multi-hop network tomography," in *Proc. of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, 2012, pp. 15–28.

[22] M. N. Mendoza *et al.*, "HMP: A hybrid monitoring platform for wireless sensor networks evaluation," *IEEE Access*, vol. 7, 2019.

[23] M. Veillette *et al.*, "CoAP management interface," IETF, Internet-Draft draft-ietf-core-comi-07, July 2019.

[24] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois, "Monitoring KPIs in synchronized FTDMA multi-hop wireless networks," in *Wireless Days (WD)*, March 2016, pp. 1–6.

[25] A. Lahmadi, A. Boeglin, and O. F. Inria, "Efficient distributed monitoring in 6LoWPAN networks," in *Proc. of the 9th International Conference on Network and Service Management (CNSM)*, Oct 2013, pp. 268–276.

[26] F. Brockners *et al.*, "Requirements for in-situ OAM," IETF, Internet-Draft draft-brockners-inband-oam-requirements-03, March 2017.

[27] J. Hyun, N. Van Tu, and J. W. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2018.

[28] M. Machacek, "Network monitoring for cloud," in *Proceedings of Optical Fiber Communications Conference and Exposition (OFC)*, 2018.

[29] F. Brockners *et al.*, "Data fields for in-situ OAM," IETF, Internet-Draft draft-ietf-ippm-ioam-data-06, July 2019.

[30] T. Watteyne, M. Palattella, and L. Grieco, "Using IEEE 802.15.4e time-slotted channel hopping (TSCH) in the internet of things (IoT): Problem statement," IETF, RFC 7554, May 2015.

[31] "IEEE standard for low-rate wireless personal area networks (WPANs)," IEEE, Standard 802.15.4, 2015.

[32] T. Kivinen and P. Kinney, "IEEE 802.15.4 information element for the IETF," RFC 8137, May 2017.

[33] R. Alfvin, "802.15.4 ANA database," January 2019. [Online]. Available: https://mentor.ieee.org/802.15/documents?is_dcn=257&is_group=0000

[34] P. Kinney, "IE characteristic table," IEEE, Tech. Rep., Aug. 2015. [Online]. Available: https://mentor.ieee.org/802.15/dcn/15/15-15-0090-08-0mag-ie-table.docx

[35] R. Struik, "Formal specification of the CCM* mode of operation, IEEE P802.15 working group for wireless personal area networks (WPANs)," September 2005.

[36] S. Duquennoy *et al.* Contiki-NG: The OS for next generation IoT devices. [Online]. Available: https://github.com/contiki-ng/contiki-ng

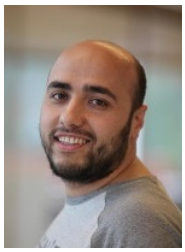
[37] S. Duquennoy, A. Elsts, B. A. Nahas, and G. Oikonomo, "TSCH and 6TiSCH for Contiki: Challenges, design and evaluation," in *Proc. of the 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2017.

[38] A. Ludovici *et al.*, "Implementation and evaluation of the enhanced header compression (IPHC) for 6LoWPAN," in *The Internet of the Future*, M. Oliver and S. Sallent, Eds. Berlin, Heidelberg: Springer, 2009, pp. 168–177.

[39] Texas Instruments, "CC2538 powerful wireless microcontroller soc for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee applications," Tech. Rep. [Online]. Available: http://www.ti.com/product/CC2538#

[40] G. Daneels *et al.*, "Accurate energy consumption modeling of IEEE 802.15.4e TSCH using dual-band OpenMote hardware," *Sensors*, 2018.

[41] Wireless testlab and officelab. [Online]. Available: https://doc.ilabt.imec.be/ilabt/wilab/



Abdulkadir Karaagac received the master's degree in Communication Systems from EPFL, Lausanne, Switzerland, in 2013. He is currently pursuing a Ph.D. degree at the Internet and Data Laboratory (IDLAB) Research Group, Department of Information Technology, Ghent University, Ghent, Belgium. His current research interests include the Internet of Things (IoT) and Wireless Networks, with a focus on mobile and wireless connectivity, robust wireless communication, network diagnosis, and interoperability.

Eli De Poorter is a professor at Ghent University, where he is coordinator of several national and international projects. Since 2017, he is also affiliated with the IMEC research institute. His main research interests include wireless network protocols, network architectures, wireless sensor and ad hoc networks, IoT, indoor localization and self-learning networks, with a strong focus on experimental wireless testbed research. He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences.

Jeroen Hoebeke is a professor in the Internet Technology and Data Science Lab of Ghent University and imec. He is conducting research on wireless communication solutions for the Internet of Things. On top, he investigates how open standards can be used to roll out connected devices and easily integrate them in IoT applications. He has been active in several IoT domains such as smart building, logistics, healthcare, industry 4.0, etc. and is author or co-author of more than 100 publications in international journals or conference proceedings.