

## Article

# Distributed Reprogramming on the Edge: A New Collaborative Code Dissemination Strategy for IoT

Gabriel Mujica \*  and Jorge Portilla 

Centro de Electrónica Industrial, Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain; jorge.portilla@upm.es

\* Correspondence: gabriel.mujica@upm.es; Tel.: +34-910-676-944

Received: 30 December 2018; Accepted: 21 February 2019; Published: 28 February 2019

**Abstract:** The ongoing era of the Internet of Things is opening up new opportunities towards the integration and interoperation of heterogeneous technologies at different abstraction layers, going from the so-called Edge Computing up to the Cloud and IoT Data Analytics level. With this evolution process the issue of efficient remote reprogramming on the Edge and the Extreme Edge deployments is becoming accentuated, as the amount and diversity of embedded sensing platforms is getting larger. To take advantage of the participation of heterogeneous devices and their in-field dynamic collaboration, in this work a new distributed code dissemination strategy for Edge node reprogramming is proposed, so as to efficiently support the functional reconfiguration, optimization and updating of sensor devices. It combines a partial reprogramming engine integrated into a modular sensor node architecture, with a smart IoT wearable platform for implementing the field collaborative framework. Results show that the proposed solution outperforms other traditional centric dissemination strategies, particularly when expanding the network reprogramming diversity and scale, which is an increasingly common feature in the IoT device deployments and maintenance.

**Keywords:** WSN remote reprogramming; code dissemination; Internet of Things deployment; wearable devices; edge computing; wireless sensor networks; hardware-software co-design

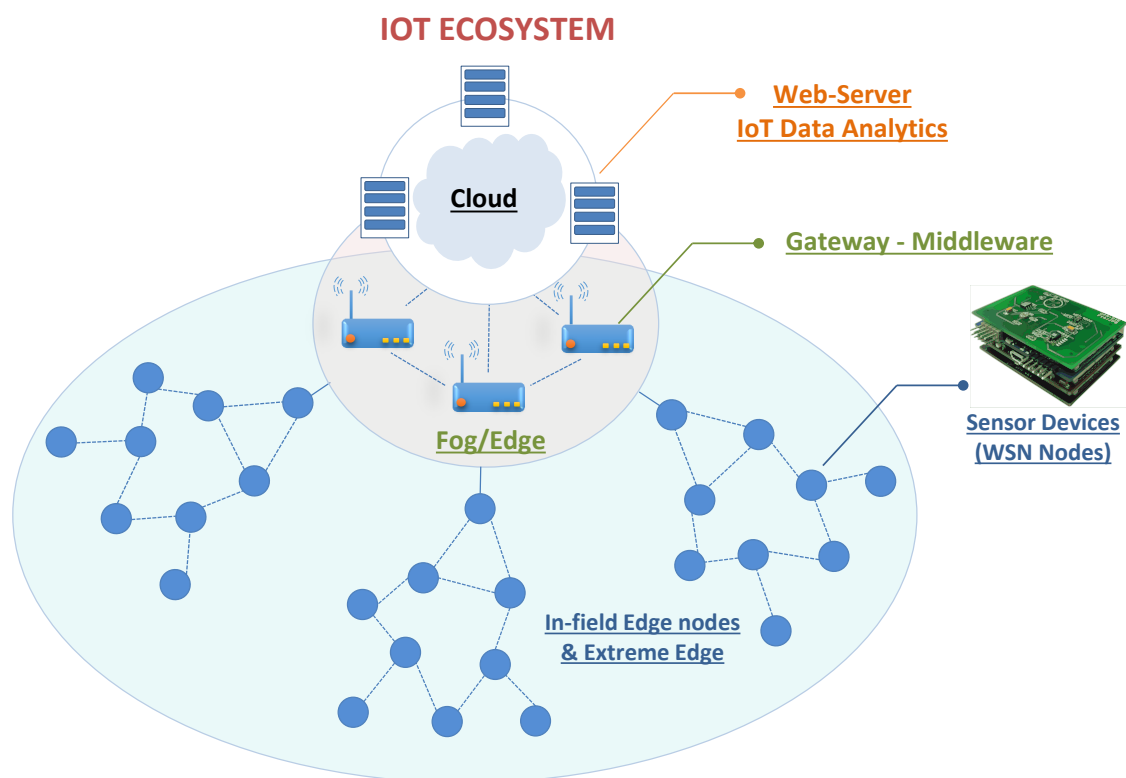
## 1. Introduction

The IoT (Internet of Things) is attracting attention from both industry and academia due to the broad spectrum of heterogeneous research areas that are involved in the implementation of smart environments. The concept of smart and sustainable cities is quickly becoming more pragmatic thanks to the application of IoT technologies in real scenarios [1]. Important contexts such as Industry 4.0 and intelligent manufacturing are key areas to be benefited from the ongoing revolution of the IoT [2].

This significant growing process also implies new challenges, not only because of the complexity of integrating different end-to-end technologies to achieve seamless interoperability and cross-connectivity among participant agents, but also the difficulties involved in properly handling the number of devices that are thought to coexist in such intelligent ecosystems. For instance, authors in [3] discuss the approach of service model to abstract the hardware and software resources into a set of services for their provision to remote consumers in Smart City contexts. They highlight the importance of data acquisition, remote collection and reusability of sensing capabilities as services. The IoT paradigm is conceived based on three major abstraction layers as depicted in Figure 1: cloud, fog and edge computing [4], and the in-field Edge node deployment plus what is known as the Extreme Edge [5], which corresponds to the physical realization and distribution of the sensor networks. The latest exhibits important challenges due to the inherent constraints of its involved technologies: limited resources, low power, large-scale distribution, self-diagnosis, autonomy and long-term operability.

There are different aspects that take part in the realization of the Edge and Extreme Edge layers. On one hand, long-term lifetime designs based on energy harvesting strategies represent an important

ongoing topic of interest for the research community to guarantee the operation of deployed devices for long periods of time [6]. The authors in [7] highlight the importance of energy efficiency in WSN (Wireless Sensor Networks), presenting the suitability of different harvesting techniques to exploit the idea of extracting energy from the deployed environment. This is to be combined on the other hand with the implementation of scalable, modular, interoperable, heterogeneous, and reconfigurable/reprogrammable embedded systems, so that the confluence between power autonomy and functional autonomy can be reached. The runtime dynamic interaction with the in-field devices encompasses the ability to add, update, replace or modify functional components and even their behavior in an efficient yet feasible manner, considering the impact (minimum or desired) in the local/overall system the device is part of.



**Figure 1.** General view of the IoT Ecosystem composed of three main abstraction layers: Cloud, Fog/Edge, In-field Edge node deployment and Extreme Edge.

Traditional WSN can be seen as an implementation of the in-field Edge and the Extreme Edge layer, although today's hardware technology may change the way embedded devices are conceived to create the wireless sensor nodes, compared to fifteen or twenty years ago [8]. An added complexity to the IoT in-field Edge and Extreme Edge layer is associated with a great level of platform heterogeneity both at hardware and software levels, which may ultimately impose the scope and limitations of the repair/reconfiguration actions to be executed on the participant elements. In [9] authors highlighted some of the key technical issues for the realization and maintenance of real sensor network deployments, such as heterogeneity, scale, connectivity, self-management, and mobility. In [10] these research challenges are emphasized in combination with the importance of implementing resource management and distributed techniques and protocols, for tackling the amount of participant devices and their associated data in IoT.

This poses several questions: How can such an amount of heterogeneous devices be managed to ensure long-term operability, (re)usability, capacity of being updated, optimized and/or replaced? In this article some of these questions are tackled by analyzing and proposing a new distributed code dissemination and reprogramming scheme for networked embedded systems within the context of IoT

smart applications, with a particular focus on large-scale and functional diversity in sensor network deployments. While most of the traditional proposals of remote reprogramming on sensor nodes were focused on performing homogeneous reprogramming (that is, all the nodes or groups of them shall receive the same code update) here the granularity of the remote reprogramming is broadened not only to the possibility of partial modifications on the sensor nodes themselves (the so-called partial reprogramming), but also in terms of the local diversity of them. This means efficient runtime selective actions according to the type of role the nodes play in the IoT deployment, in addition to their intrinsic nature for which they are participating and collaborating in the target application.

This work especially takes into consideration the integration of smart mobile elements (in this case wearable devices) with deployed sensor nodes, and their dynamic interaction and collaboration in IoT scenarios. This approach is thought to also empower cooperative computing strategies on the Edge. The main contributions of this work are summarized as follows:

- An in-field distributed code dissemination strategy for collaborative reprogramming among heterogeneous devices on the IoT field deployments. A particular focus on dynamic IoT smart contexts where the cooperation between mobile wearable objects and wireless sensor nodes is fostered.
- The combination of the proposed dissemination approach with an efficient partial reprogramming mechanism to produce a fine-grained node and network updating, so that a long-term functional operation and adaptation can be guaranteed.
- The realization of the distributed techniques by means of an extended IoT hardware-software integration framework, which relies on two modular and flexible embedded architectures.
- A comparative analysis of the proposed approach against traditional dissemination solutions to show the main benefits and challenges of these types of distributed strategies for the present and future of IoT Edge and Extreme Edge deployments.

The rest of the article is organized as follows. Section 2 is devoted to the analysis of traditional techniques and new trends of code/data dissemination in WSN, whereas in Section 3 the proposed in-field dynamic dissemination strategy is described in detail. Then, Section 4 introduces the baseline architecture of the heterogeneous platform for the support of IoT collaborative reprogramming, while in Section 5 an experimental comparison of several remote reprogramming schemes is presented. Finally, conclusions and future works are highlighted in Section 6.

## 2. Related Work

The remote reprogramming of wireless sensor nodes once the deployment is in its operational stage is a major concern for WSN and IoT developers, since it allows performing runtime adjustments or even strong modifications on the in-field system, without the need for decommissioning part of or the entire network [11]. The effective accomplishment of such tasks will certainly have an impact in terms of cost, lifetime and long-term functional integrity of the implemented sensor network.

In traditional WSN the remote reprogramming of sensor nodes has been studied mainly from two points of views, i.e., the type of update to be carried out and the dissemination technique to be applied. Full code reprogramming is the simplest way of performing the software modification since no particular embedded structure beyond the definition of an internal frame composer and a program bootloader is needed. However, a strong penalization in terms of reprogramming time, energy consumption and network overhead makes this approach unfeasible for WSN. Flexibility, software reusability and heterogeneity are limited both locally (no partial modifications can be made on the sensor nodes) and globally (a fine-grained modification of different diverse nodes become unrealistic). To overcome these shortcomings, differential and incremental reprogramming, loadable modules and virtual-machine-based application modification are some of the main methods to face this first issue.

Virtual-machine-based techniques promote high-level abstraction for the development of embedded software components that can subsequently be updated or replaced, without modifying the

rest of the common support structure. Maté [12] was one of the first pioneer implementations of this type of strategy for WSN and the inspiration for subsequent tiny VMs proposals. Loadable modules rely on the support of embedded operating systems such as Contiki [13] to provide the capability of linkable modules which contain references to functions and variables of the system [14]. Their access has to be resolved by using static or dynamic linking strategies. Incremental reprogramming considers a differential-based technique where mismatches between new and old versions of the software to be reprogrammed are calculated, so as to generate a delta image [15]. The latter is the piece of code that will finally be transmitted to the remote sensor node, which is the one in charge of resolving the differences before performing the corresponding reprogramming. Thus, this technique usually needs additional memory support to carry out the code comparisons. Other researchers try to have a trade-off between heterogeneity of sensor platforms and the level of updating in next-generation large-scale deployments. In [16] authors overcome the complexity of reprogramming WSN by exploiting the notion of wireless reconfiguration. Instead of implementing a particular reprogramming model, they propose the modification of key sensor policies regardless of its hardware and software architecture, so the scope of the update is at parameter level.

Efficient code delivery using the main wireless link of the sensor network has also been a research topic of interest in WSN due to the importance of updating remote nodes with a minimum overhead yet high ratio of success. Fulfilling these requirements are indeed challenging, since low-power and low-rate networks pose important problems associated with packet lost ratios (and thus reliability), resource availability, and battery degradation. Deluge [17] is one of the main reference techniques that proposed a dissemination protocol for object propagation in a reliable fashion. It particularly focuses on binary partitioning and end-to-end software composition to ensure the integrity of the program to be updated. Works such as [18] try to overcome and optimize the limitations of Deluge and similar centralized approaches, particularly related to the fact that the Deluge component had to be included in the new image to be transferred (thus increasing the overhead). To further mitigate the overhead caused by code dissemination, especially in traditional solutions based on triggering it from the base-station down to the sensor nodes, authors in [19] propose a mobile version of Deluge, which basically performs a one-hop dissemination by using a mobile base station connected to a laptop.

With the evolution of IoT and networked embedded systems, code dissemination and updating shall also emerge to distributed techniques where different heterogeneous devices can join large-scale and dynamic deployments, to exploit the collaboration and interaction among participant elements. Contributions related to the adaptation of multicast-based data transmission schemes from IP-networks [20] reaffirm the importance of efficient dissemination under resource-constrained distributed sensor networks. Other works attempt to make the dissemination process more efficient in relation to the channel conditions of the target scenario, such as in [21] where a packet size adaptation during the dissemination process is done in accordance with the link quality estimation provided by the radio module. This approach is thought to be applied in monolithic TinyOS-based deployments or as an extension of Deluge-based strategies, so both heterogeneity and in-field distributed adaptation is not accounted for. Performing the code dissemination in a distributed fashion has also been highlighted in [22], where a decentralized mechanism for WSN is introduced. The authors stress the idea that more than a single authorized actor beyond the base-station is to be considered for accomplishing an efficient yet secure code updating on the sensor nodes. Likewise, authors emphasize [23] the importance of distributed data discovery and dissemination in IoT beyond the traditional centric approaches. They particularly focus on vulnerabilities and trust issues to satisfy secure data dissemination. Some authors also focused on efficient reprogramming regarding throughput and minimum overhead, as in [24], where a network coding technique is used to optimize the amount of disseminated data as well as its integrity, although it is primarily intended to be applied for propagation of the same program image from a source node to all sensor nodes.

### 3. Proposed Distributed Strategy

As discussed in Section 2, when analyzing possible update scenarios in WSN (and extended to IoT) two main aspects may be taken into account: (a) what shall be reprogrammed/updated/replaced and (b) how the action is performed. In the first case, the configuration/code segmentation level is defined by the software architecture implemented for supporting dynamic reprogramming of particular elements in the node, while in the second case the functional diversity of the sensor devices (i.e., granularity of the network) will impact the message propagation strategy to be used.

This work makes use of the partial reprogramming strategy proposed in [25] to face code updating in sensor nodes (a), and aims at creating a distributed code dissemination scheme to carry out the reprogramming process (b). This scheme is extended to runtime reconfiguration of distributed devices, in the sense of updating or evolving functional parameters of their behavior. This means that the distribution techniques are also suitable for several forms of data/code distribution, feature that is of great importance in large-scale IoT application contexts.

The embedded reprogramming engine integrated within the proposed distributed strategy makes use of an efficient modular segment-based partial reprogramming mechanism to carry out the over-the-air code update process. It combines the definition of three main areas: The bootloader and recovery segment, which is a memory-protected component that allows receiving and managing the reprogramming composition and over-the-air updating process; a software support area, which includes the hardware-software integration platform to abstract the low-level control and development of embedded applications within the scope of the Cookie architecture; and a user dynamic area, where different and independent application segments can be reprogrammed according to the application or experimentation needs. To reduce the size of the images to be reprogrammed, the bootloader recovery segment is embedded as a fundamental part of the remote reprogramming system, so that there is no need to send it every time a reprogramming task has to be triggered. These areas are seamlessly correlated by means of a dynamic linking area, which allows transparent and independent code updating, replacement and modification of both the user dynamic area as well as the software support area (in case new controllers or specific modifications of the support functionalities are to be addressed). A whole reprogramming process is not necessary in either of these cases, thanks to the relative linking strategy applied between them. The data transfer reduction with this strategy can be up to 95% in comparison with a traditional whole reprogramming mechanism, which benefits the proposed distributed strategy in terms of the diversity of the code updating, the reprogramming time reduction, and thus the power consumption during the transferring and updating process.

The traditional code dissemination in WSN relied on the coordination-based structure of the wireless network, i.e., considering packet propagation from the root device down to the different route/sensor nodes, or flooding-based strategies [26]. These approaches would be a possible solution when considering low granularity regarding hardware and software diversity of the sensor network. In large-scale heterogeneous IoT deployments, networks are closer to mesh or hybrid structures where the data gathering is rather distributed throughout several PoI (Point of Integration) with the fog layer. Furthermore, intelligent cooperation of heterogeneous participant devices can be used to optimize the reprogramming process in field. These two concepts benefit the proposed solution.

The structure selection and dynamic evolution of a network can suite an enhanced reprogramming strategy, in accordance with its granularity. This means that how routes are dynamically created and what relationships exist among the nodes (particularly in terms of the application and software components they include) would be used to create a distributed and granular dissemination approach. The main differences rely on the concept of “relative notion”: what information is handled by the participant agents with respect to the local/overall system. In centric-based structures, such information is limited to routes towards root devices and neighbor nodes (at most). So the notion of a sensor node with respect to the network or the surrounded area is rather limited. In case of clustered-based structures, local information is distributed among cluster-heads and, depending on the routing implementation, adjacent clusters are also registered. A trade-off between memory

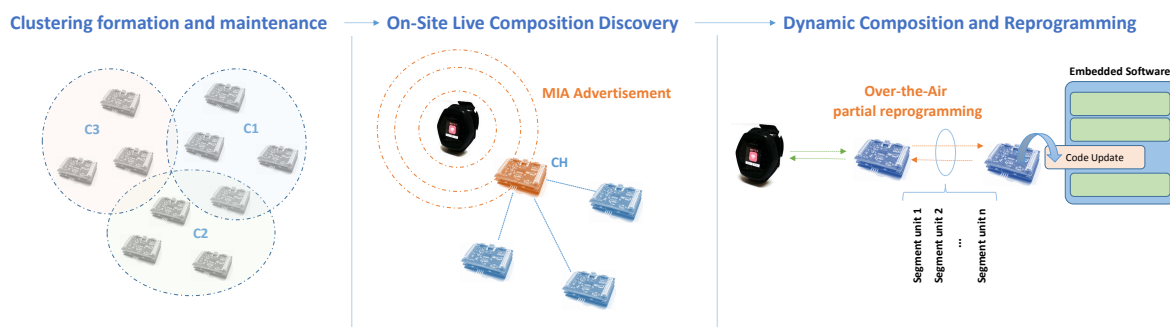


handling, communication overhead and power consumption policies usually imposes the boundaries for extending such a relative notion. This work combines an adaptive clustering technique with the inclusion of active mobile agents, in order to support distributed reprogramming with high granularity yet reduced overhead on the network. The in-field interaction between heterogeneous nodes and their relationship is exploited to produce an efficient dissemination context for reconfiguration and code updating purposes.

Thus, the proposed solution takes advantage of cluster heads to disseminate binary code or node configuration towards cluster's participants, so that the overhead can be minimized compared to network flooding or tree-based down-link packet exchange, as demonstrated subsequently. However, the great potential of such an optimization lies on making use of intelligent agents that are part of the IoT network and dynamically interacts with the cluster heads in field, in order to decide what nodes shall be updated at runtime depending on the granularity indicated by the evolution of the clusters, and according to the information contained in the CH (Cluster Head). To highlight the presence of the mobile intelligent agents, other approaches without cluster-based routing are also shown so as to compare the performance of using both options in contraposition to the ones mentioned before.

### 3.1. Phases and General Structure

The MDS (Mobile code Dissemination Strategy) is composed of three main phases, in which the dynamic clustering formation performed by the deployed nodes is combined with the live discovery and dissemination composition featured by the mobile elements, as represented in Figure 2, and described as follows:



**Figure 2.** General structure of the proposed distributed reprogramming strategy and representation of the different involved phases.

#### 3.1.1. Clustering Formation and Maintenance

This stage is related to the networking structure upon which data/code dissemination is performed. A cluster-based routing approach with dynamic CH selection according to energy conservation + sensor node composition underlies the proposed distributed system. The cluster formation starts with a discovery process where sensor nodes evaluate their surrounding area to decide whether they are candidates to form a cluster and how they participate on it. Four types of nodes are distinguished: IN (Isolated Node), SN (participant Sensor Node), CH and LN (Linker Node). The CH selection is carried out by using a composed quality metric that serves as an evolving weighing indicator to dynamically change the role of the involved nodes.

Targeting energy-efficient deployments, the battery degradation and computational duty-cycle are considered to obtain the weighing indicator, which will be reinforced with an additional parameter to evaluate the suitability of a node to take part of a cluster. This parameter, referred to as SCm (Sensor Composition), allows detection and local evolution of the network granularity, so that the code/configuration updating can be enhanced (thus data fusion context as well). SCm is determined by the type of sensing capabilities of a node in the network, and the composition of partial reprogramming slots to be updated in a node. Every sensor node has a software composition identifier which allows

distinguishing the type of device in terms of overall application and sensing capabilities. The specific parameters that can be configured for a given type of node define a second level of sensor composition, in accordance with its realization. Then, for a given type of node a particular segment composition regarding the partial reprogramming architecture defines the specific functionalities that a sensor node can execute. These three composition levels will ultimately define the depth of the SCm and therefore the platform diversity of the clusters to be formed. Such a diversity will delimit the granularity of the reprogramming within local areas. The CH registers the SNs that are part of the cluster in addition to their composition and weighing indicator, so that the tracking of the local diversities can dynamically be updated.

### 3.1.2. On-Site Live Composition Discovery

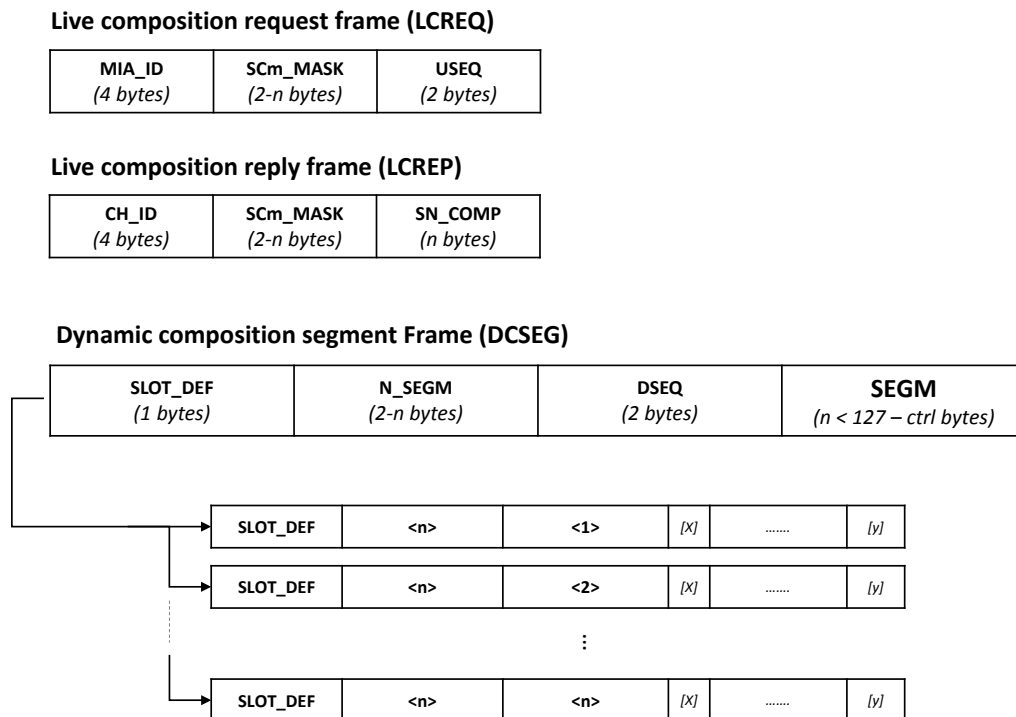
In this phase the MIA (Mobile Intelligent Agent) takes part in the distributed strategy as a mobile Edge node that dynamically participates in the sensor network. Their selection is inherent to the heterogeneous nature of the WSN, that is, to the intrinsic nature of the type of participant device. In this case, the smart wearable nodes act as the MIA so as to provide the in-field distributed reprogramming strategy with the support of the CHs. They accommodate the extended functionalities to underpin and contribute to the dynamic configuration and code updating process of the involved sensor nodes. The MIA triggers an interactive discovery of cluster composition in field to dynamically build an optimized reprogramming array. By combining the information contained in every CH with live composition packets performed by the mobile device, the overhead of the network is considerably reduced. The mobile agent advertises its presence by emitting a live composition packet that is only replied by surrounding CHs in the target deployment area, with the clustering composition as well as the weighing indicator of the participant nodes.

Thus, the MIAs can have a twofold supporting role in the wireless network. On one hand they participate as collaborative sensor nodes in the IoT deployment, so that they provide their sensing and processing capabilities to the overall ecosystem, while on the other hand they act as mobile seamless reprogramming agents. To provide the latter capability, MIAs include a repository of updating units that correspond to the type of reconfiguration, replacement, and/or reprogramming to be performed, i.e., partial reprogramming slots or an array of reconfiguration objects. This repository is updated whenever the MIAs directly interact with the fog layer, which can happen on-site or off-site. In this way, the power budget of the MIAs does not impose a primary limitation in comparison with the deployed sensor nodes, due to their dynamic interaction with the in-field Edge network and due to the fact that their power supply integrity relies on the associated mobile participant and their subsequent periodic updating.

Figure 3 shows the structure of the live composition packet, which includes:

- ID (Identifier) of the MIA.
- SCm Mask that allows advertising the type of update that can be performed according to the current repository of the participant MIA.
- USEQ (Update Sequence Number) related to the associated repository entry, so that nodes can evaluate if they have to be updated or, instead, they are running a proper version of their sensor composition. To minimize the packet overhead in this stage, the CH performs a first internal evaluation (according to the information of the latest clustering refresh) to decide what nodes have at least a former version of the sensor composition, so that they will be included in the following stage. In case an equal or later version is detected, such nodes will not take part of the third phase.

The live composition reply includes the participant sensors on the cluster and their respective SCm, an update flag indicating the status of the node at the latest refresh and, an adjacent clustering array containing the parameters of border clusters (if available). This information allows supporting a finer definition of a well-delimited implicit coverage area to dynamically lead and perform the in-field reprogramming process.



**Figure 3.** Control packet frames for performing the on-site dynamic code dissemination strategy.

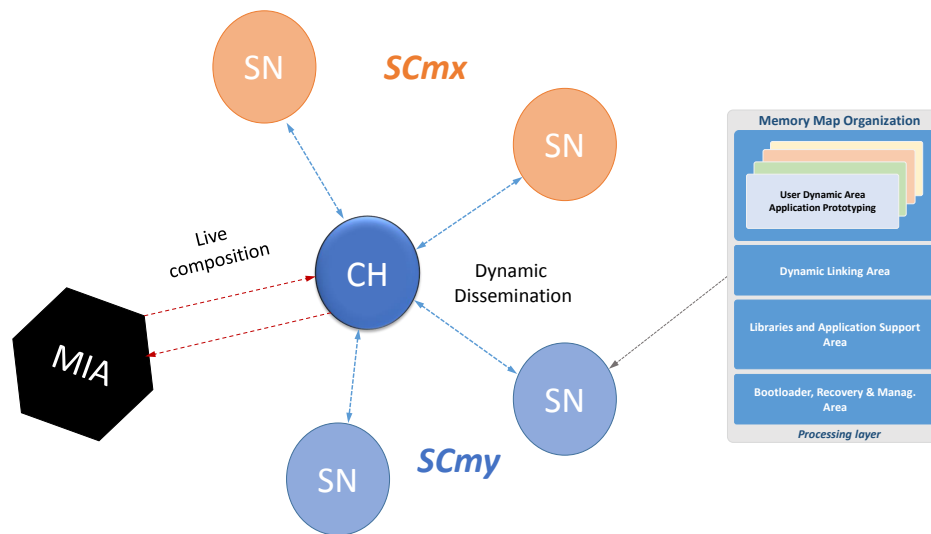
### 3.1.3. Dynamic Composition Update

Once the granularity of the cluster and the type of update to be carried out on selected participant nodes are notified by the CH, the MIA activates the on-site node reprogramming process, which is supported by a binary segmentation mechanism to conform the code update to be sent through the radio communication module of the sensor devices. As shown in Figure 3, the first packets include a control field that designates the slot/configuration to be updated, a code composition value to delimit the code dissemination segments to be transmitted, and a Dissemination Sequence number to guarantee end-to-end code integrity.

Then, the CH disseminates the code update throughout the cluster in accordance with the selected participant nodes. In case of ‘coarse granularity in cluster’, the CH reprograms with a broadcasting scheme where sensor nodes subsequently inform the final composition of the received segments, so that a local repair mechanism can be executed if needed. In case of ‘mid granularity in cluster’, an announcing control packet is first distributed to the participant nodes including a composition mask that allows devices to decide whether they shall participate in the updating process. Then, nodes out of the composition enable low-power mode to reduce power and processing expenses, and thereafter the code dissemination process is executed. The announcing mechanism is repeated until all the different composition updates are successfully accomplished.

This process is represented in Figure 4, where the different phases of the dynamic composition update are highlighted, and how the MIAs play a significant role to generate an efficient and optimized dissemination strategy by taking advantage of their participation in the heterogeneous network. As the number of MIAs and their diversity increase within both local and overall network, the effectiveness of the proposed solution will have an important energy and cost improvement on the deployment lifetime, so the functional operation of the devices can be extended. The figure also shows the general structure of the partial reprogramming scheme that provides the underlying support for lightweight application reprogramming of different pieces of software blocks, as commented before.





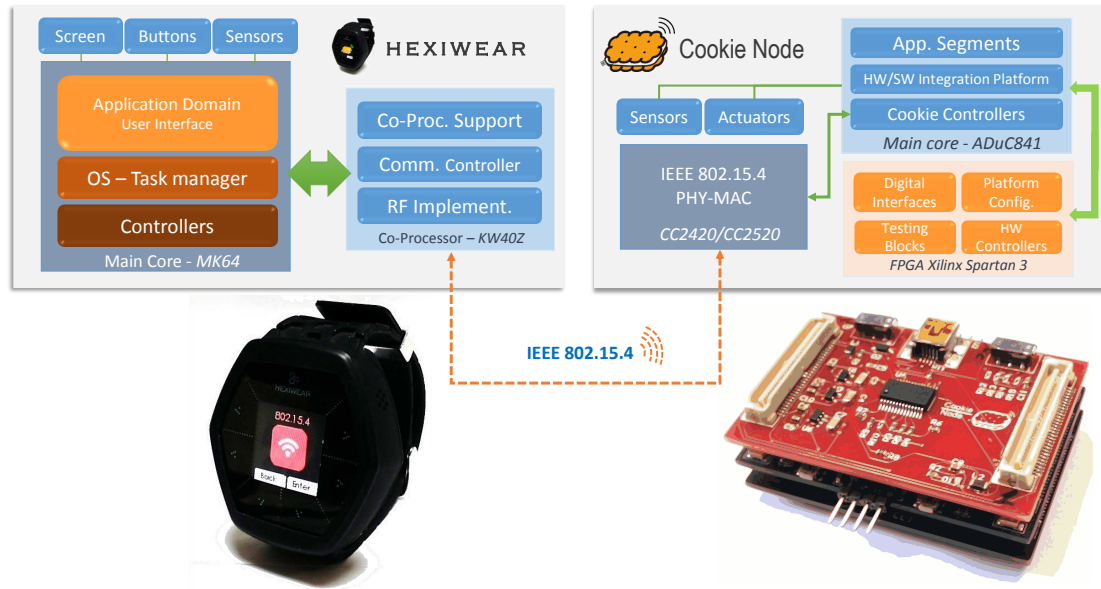
**Figure 4.** Representation of the different phases within the Mobile code Dissemination Strategy (MDS), considering a cluster with different SCms. The partial reprogramming structure embedded in the sensor nodes is also depicted.

#### 4. Baseline Architecture

The implementation of the proposed distributed reprogramming strategy is conceived under the basis of a heterogeneous IoT hardware-software architecture [27], which is composed of a modular and flexible WSN platform and a smart wearable device for IoT. On one hand, the Cookie device [28] allows fast prototyping based on the inclusion of different interconnected layers that can be customized according to the requirements of the target application, and the reusability of hardware components is indeed promoted. The modular structure of the Cookies fosters interoperability of diverse IoT technologies, since different processing elements, communication protocols, energy and sensing capabilities can be integrated following the available top-down interfaces for transferring power and data among layers. In this way, the Cookie architecture provides the basis for having seamless integration with (1) the target application context, by designing the particular sensor layer adapted to the environment and sensing coverage requirements but reusing any of the already available processing elements that can address specifications (fast computing, ultra-low power modes, wake-up policies, digital signal processing); (2) the fog layer, by using a heterogeneous stack that may include several forms of communication interfaces and processing capabilities, so that an end-to-end interaction with the Cloud (when needed) can be possible; and (3) with various IoT devices mainly by means of standard relationships, such as well-established protocols for cross-connectivity among participant elements on the deployment. As mentioned before, for the proposed distributed dissemination strategy, the Cookie nodes make use of a partial reprogramming scheme that allows on one hand efficient reprogramming of independent application-wise segments defined by users as well as support libraries of the Cookie integration framework, and on the other hand reconfiguration of device parameters to modify the behavior of the running top level application.

On the other hand, the Hexiwear [29] provides a development environment for exploiting the characteristics of a sensing wearable device towards IoT intelligent scenarios. The Hexiwear architecture integrates two NXP Kinetis microcontrollers [30] and a combination of sensors plus graphical/touch interfaces to ease the interaction of the user with the smart wearable device. One of the MCU includes an ARM Cortex-M4 (1M Flash, 256K SRAM) that supports the integration of real-time embedded Operating Systems for software and resource management. The second MCU (ARM Cortex-M0+ core) includes the communication technologies that favor the integration of the wearable device with the fog and edge layers. In particular, the proposed distributed strategies make use of the Hexiwear IEEE 802.15.4 support to have a full integration with deployable sensor nodes

(the Cookies include a CC2420 as well as a CC2520 [31]-based communication layer within the available modular architecture). Efforts have been made to have a full communication among these platforms, not only from the point of view of direct and seamless connectivity between devices, but also from the high-level development of embedded applications based on this hybrid solution. Figure 5 shows both platforms and the customization of the Hexiwear to interact with the Cookie nodes in field.



**Figure 5.** Heterogeneous HW-SW platforms that support the IoT collaborative strategies on the in-field Edge deployment: The modular Cookie node and the Hexiwear smart wearable device.

## 5. Experimental Evaluation and Comparative Analysis

Considering an IoT network deployment composed of  $N$  nodes, where a cluster-based routing strategy is used to communicate among participant nodes and with the fog layer,  $CH_n$  is the Cluster Head of cluster  $n$ :

$$C_n(CH, S, LN) = CH_n + \sum_{i=1}^m S_i + \sum_{j=1}^{l < m} LN_j \quad (1)$$

$S_i$  represents the participant nodes in cluster  $n$ ,  $LN_j$  is the linker node between two adjacent clusters. Let's also consider a code dissemination strategy based on data partitioning where:

$$f_{seg}(p) = \sum_{z=1}^q p_z \quad (2)$$

$f_{seg}(p)$  represents the code to be disseminated throughout the network, and  $p$  refers to the segment unit that is part of the code update to be disseminated. Hence, if the code update to be transferred is greater than the maximum size of the unit of fragmentation, then  $q > 1$ .

First of all, consider a network composed of 30 nodes and with the distribution represented in Figure 6, where three different remote communication strategies for the dissemination process are considered. First, a flooding-based flat mechanism which is triggered by the root point (PoI), and where the surrounding nodes retransmit the dissemination control frame to the neighbor nodes so that the reprogramming stage is spread throughout the mesh; secondly a tree-based strategy by using a combination of the ETX (Expected Transmission Count) metric concept [32] with the computation of the RSSI (Received Signal Strength Indicator) and a correlation parameter with the LQI (Link Quality Indicator) provided by the IEEE 802.15.4 CC2520 radio communication module (as expressed in [31]), in which the creation of the routes are performed by considering in every node the smallest accumulated

ETX towards the PoI (where  $ETX_n = ETX_p + ETX_{p-n}$ ) so that the most optimized tree in terms of a trade-off between number of hops and quality of the involved link is accounted for; and third the described cluster-based routing strategy with the dynamic selection of the CH that will support the code dissemination process. For the simplest case: (1) a single point of integration with the fog layer; (2) two-hop distance between clusters (single-LN based array between CHs for simplicity); (3)  $q = 1$  (one segment corresponds to a unit of fragmentation, which is considered 127 bytes in this case); and (4) the network granularity is the minimum (homogeneity of nodes), the communication overhead for reprogramming the network results in a reduction of 53% in comparison with only applying a clustering strategy without the support of the proposed distributed technique.

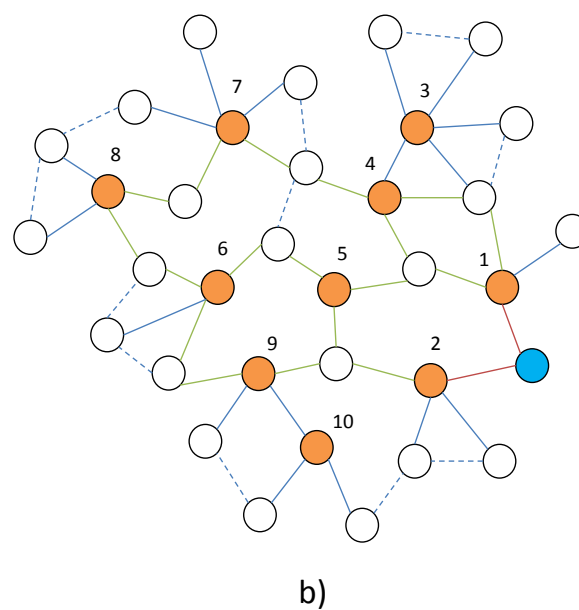
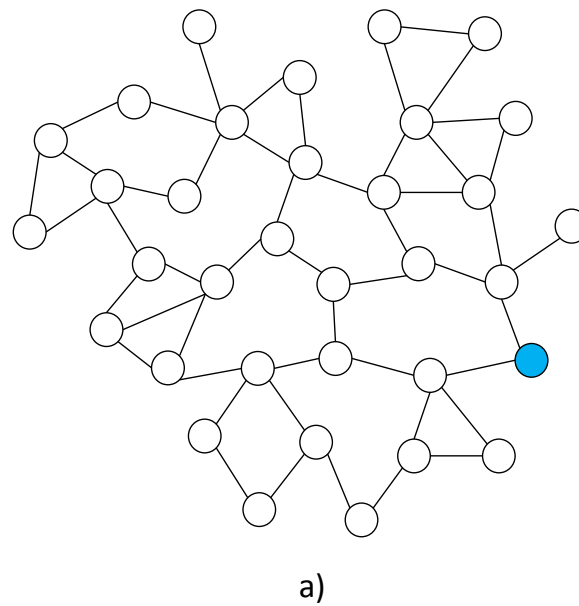
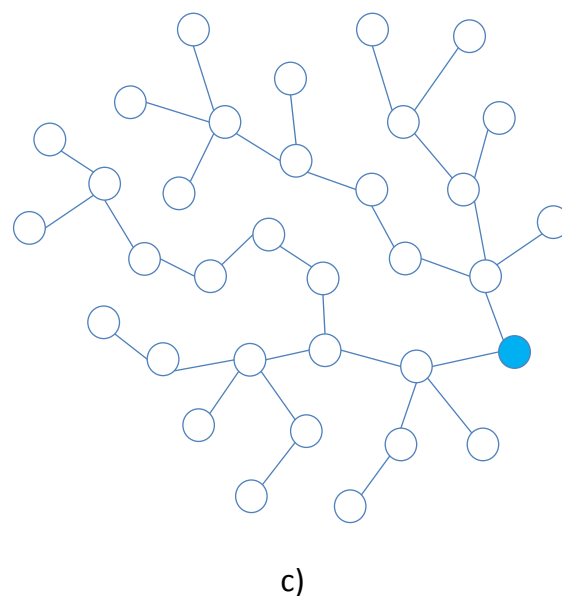


Figure 6. Cont.



**Figure 6.** Structural comparison of dissemination strategies based on the network topology: (a) Flooding (b) Cluster-based (c) Tree-based.

### 5.1. Experiments Setup and Parameters

To further analyze these results, in Table 1 a summary of the main considered parameters for performing the experimental comparison of the different dissemination scenarios in the target network is depicted.

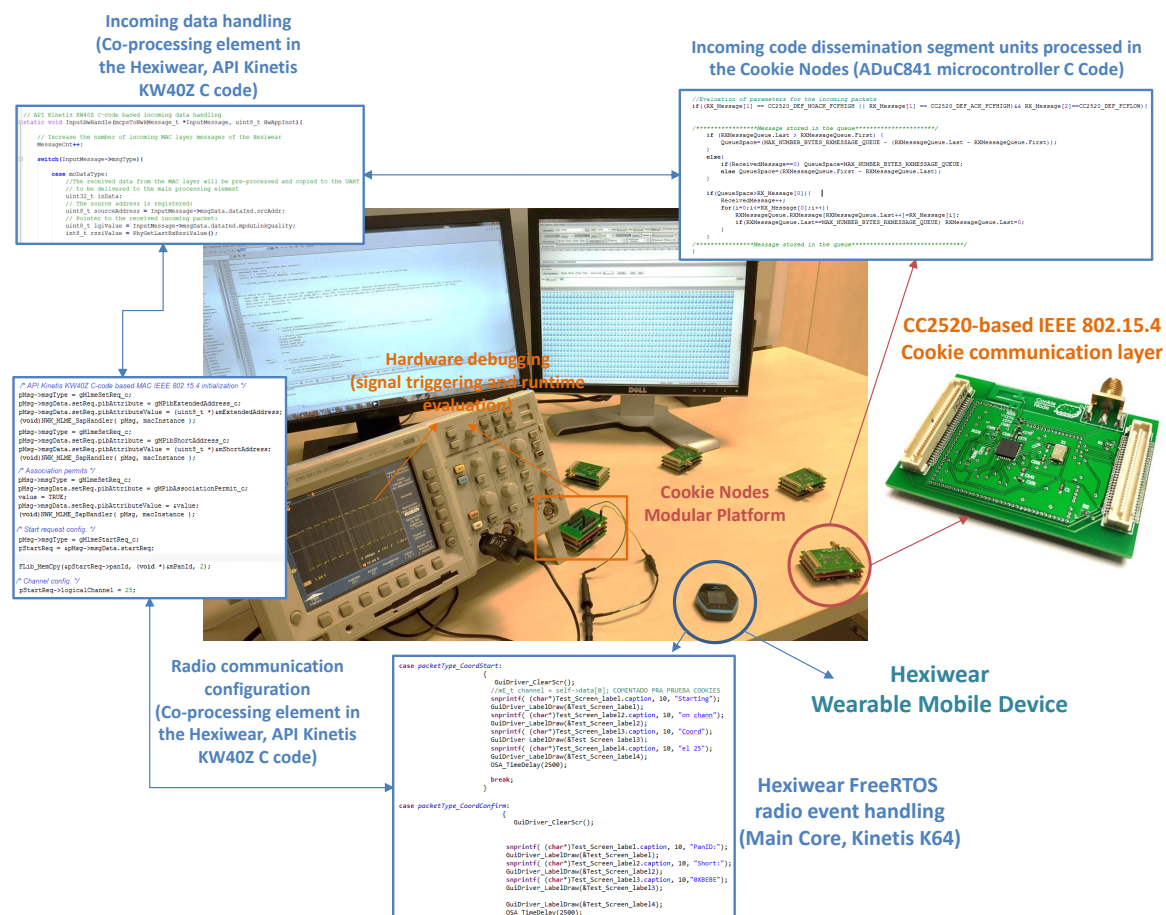
**Table 1.** Main parameters for comparing the dissemination strategies.

FEATURE	VALUE
Dissemination scheme	Flooding Clustering Tree-based Distributed MIAs
Network granularity	1, 2
Segmentation level	1 to 10
Points of Integration with the Fog (PoI, Roots-GW)	1, 2

The experimental setup for the target network is composed of a C-language-based simulation carried out with the inclusion of a connectivity correlation matrix ( $WL$ ) that serves as the baseline structure to form the wireless network, from which the different paths to the participant nodes are computed.  $WL$  primarily contains the information of the RSSI values for the different possible wireless links throughout the network, based on experimental measurements performed for the CC2520-based Cookie modular node, whose characterization allows the application of an adapted Log-Normal Shadowing Model [33] considering its parameters' adaptation.

From that, the correlation values for obtaining link quality metrics are produced, so that the correlation matrix is fed with the additional parameters to further calculate the multi-hop paths in accordance with the aforementioned topology mechanisms. Additionally, in order to verify the correct operation of the algorithm on the real sensor nodes, the simulation of the target network is combined with the actual operation of one of the clusters (thus promoting a hardware-in-the-loop like simulation scenario) as shown in Figure 7, which allowed confirming the effectiveness of the proposed distributed reprogramming process from the MIA to the participant nodes with the support of the CH.

As pointed out in Section 4, the implementation of the MIAs is performed by using the smart wearable devices and the modular Cookie platform. The latter includes an 8051-based microcontroller from Analog Devices, the ADuC841 [34], which runs the hardware-software integration platform [28] (entirely programmed in C language) considering the partial reprogramming scheme for the distributed dissemination process (including the embedded bootloader and recovery area for code update management). The main core of the Hexiwear platform runs a FreeRTOS operating system [35] for performing the general management of the device, and the co-processing element (the Kinetis KW40Z) is in charge of the management of the radio communication transceiver and the associated events and packets triggering of the IEEE 802.15.4 stack, which functionalities are programmed in C language. The programming scheme and functional flow of the heterogeneous hardware-software platform is shown in Figure 7, where an overall view of the different software elements that take part in the actual embedded devices is highlighted. For the real experimental verification, the new CC2520-based Cookie communication layer is used to carry out the wireless communication, whose design and physical implementation is also shown in Figure 7.



**Figure 7.** Experiment setup for the verification of the proposed system on the heterogeneous IoT HW-SW platform, composed of the Cookie modular nodes and the Hexiwear smart wearable device.

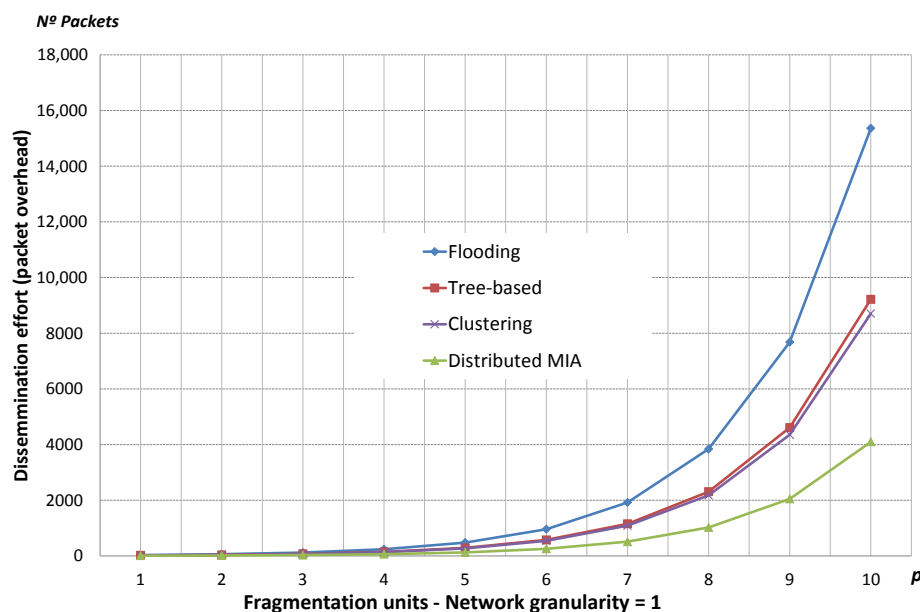
According to the parameters depicted in Table 1, the dissemination effort is computed for every dissemination scheme taking into account firstly a network granularity = 1, which means that all the nodes will be updated with the same reprogramming code, and the segmentation units is varied from 1 to 10. For every single case, the number of packets to reach the intended sensor nodes to be reprogrammed is registered, so as to compare the network overhead for every dissemination scheme. To accomplish this, the PoI triggers the dissemination process for the first three strategies, whereas for the latest case this task is executed from the MIAs, in line with the proposed distributed strategy. Then



the process is repeated by increasing the network granularity, in order to analyze how the diversity of the network deployment affects the reprogramming effort for every case. The accumulated number of packets is computed for every fragmentation unit case to compare the dissemination effort of the different dissemination strategies against the network granularity and code update composition. As also shown in Table 1, the inclusion of additional PoIs for the clustering approach is considered as well, in order to verify the benefits of increasing the number of root elements to be interfaced with the fog layer, within the context of the code dissemination process.

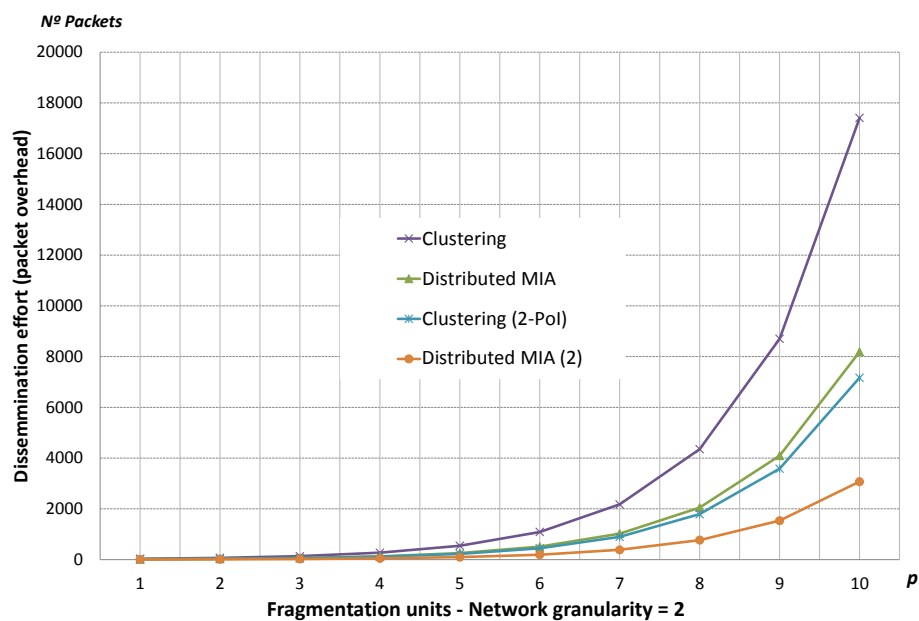
## 5.2. Results Analysis

The evolution of the code dissemination strategies according to these conditions is shown in Figures 8 and 9. As expected, the flooding technique obtained worst results in comparison with the rest, while the proposed distributed technique outperforms the others with significant differences. On the other hand, clustering provides important benefits as the number of participant nodes, PoI, and granularity-based CH selection increases, since the distributed strategy takes more advantage of the CH information to better interface the smart wearable devices during the in-field reprogramming process. This represents a 90% of improvement with respect to the highest case of overhead. Figure 9 shows the cluster-based and MIA-based strategies, which provides better results when increasing the PoI, MIAs and network granularity level (82% of reduction with respect to the highest case of overhead, i.e., with a single PoI).



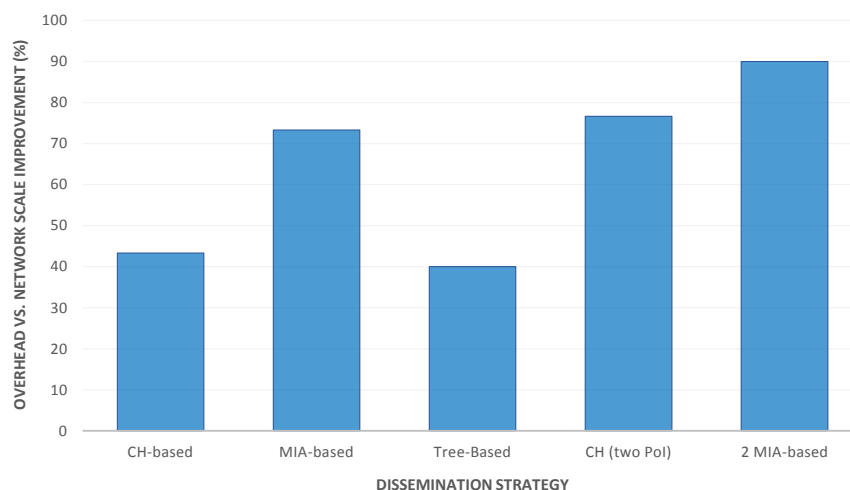
**Figure 8.** Results comparison of the proposed dissemination vs. the other strategies according to a network granularity equal to 1, showing the packet overhead tendency when the fragmentation units are increased.

It shall also be noted that an extreme position of the root with respect to the network deployment has been adopted. It means that the distribution of the PoI can also increase the performance of the clustering technique, as well as a higher number of PoI and participant mobile agents. This observation leads to highlight another important contribution of the proposed strategy. Since the code dissemination and on-site reprogramming is performed in a distributed manner, the overall energy dissipation during the process is equilibrated in comparison with centric approaches, where nodes close to the root/gateway devices have a greater participation. This is particularly noticeable when increasing the granularity of the network, i.e., the diversity of nodes to be reprogrammed is much higher.



**Figure 9.** Results comparison of the cluster-based strategies when increasing the PoI, MIA and network granularity equal to 2, showing the packet overhead tendency when the fragmentation units are increased.

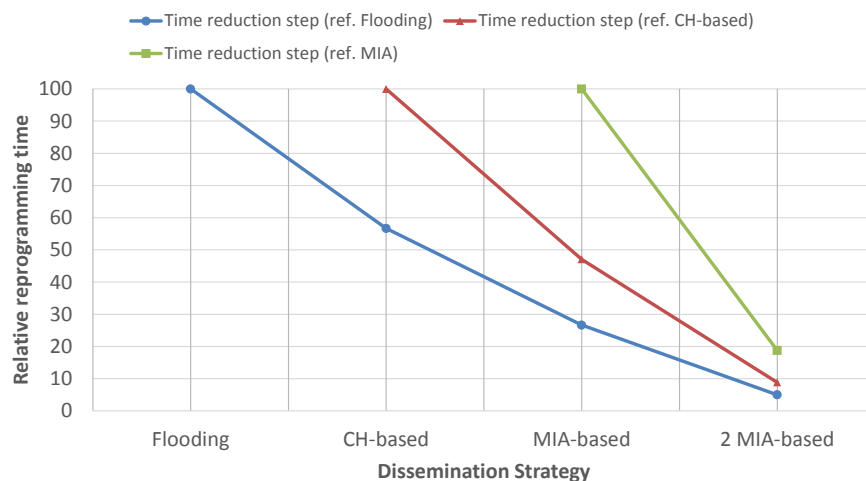
On the other hand, Figure 10 shows the results regarding the relationship between the network scale and the network overhead when applying the compared dissemination strategies, normalized with respect to the maximum value (worst-case scenario, which corresponds to the flooding technique), and for a granularity = 2, and segmentation = 10. It can be seen that the MIA-based strategy provides clear optimization results in comparison with the rest of the techniques, which is a crucial indicator when the size of the network becomes bigger. Moreover, with the support of an increased number of distributed MIAs throughout the deployment working as both mobile sensing devices and code dissemination agents, the benefits for the total long-term operation of the sensor network is noticeable.



**Figure 10.** Comparative results relating network overhead and network scale, for the different applied dissemination strategies. The percentage of relative overhead against scale is shown for every involved strategy.

Regarding the reprogramming time metric, Figure 11 depicts the comparative outcomes of the analyzed dissemination mechanisms, where the relative time reduction results are shown. The blue curve represents the evolution of the reprogramming time with respect to the worst case, where a

pronounced improvement tendency is achieved as the number of PoI and MIAs that take part on the reprogramming process grows. The 2-MIA-based experiment has the furthest time reduction, a total of 95%, whereas a 91.2% (red curve) and 81.3% (green curve) of relative optimization compared to 1-MIA-based and CH-based are obtained, respectively. The outcome data and the comparative values of the different dissemination strategies can be found in [36].



**Figure 11.** Reprogramming time curves representing the reduction tendencies among the different analyzed dissemination techniques. The timing improvements are compared by overlapping every strategy with the corresponding reference outcome.

## 6. Conclusions and Future Work

In this work, a new distributed code dissemination strategy based on heterogeneous IoT Edge platforms has been proposed for in-field collaborative network reprogramming, so that a more efficient and selective update of participant sensor nodes can be performed. The comparative analysis of different reprogramming schemes shows that the proposed MIA-based strategy can make a great improvement on the network granularity support, the packet overhead, the reprogramming time and thus power consumption balance, particularly when the diversity of sensor composition and the deployment heterogeneity grows. The future of large-scale Internet of Things deployments makes the implementation of distributed and collaborative reprogramming techniques a significant aspect to be faced, since the technology is pushing towards the real operation of a great amount of devices. Taking advantage of their interaction and collaboration, the way edge computing supports the overall IoT ecosystem will indeed evolve quickly. In this direction, one of the ongoing works relates to the analysis of a dynamically-spinning based composition discovery technique to apply the MIA-based reprogramming in flat/mesh networks, so that a comparison with the proposed strategy can lead to provide support for additional network topologies. Exploiting the in-field interaction among the MIAs themselves can produce important benefits so as to create a further mobile distributed edge computing scenario. In this way, efforts are being made to strengthen the dynamic on-site collaboration among heterogeneous platforms and participant agents. In addition to this, the exploration of hybrid structures where tree-based algorithms are integrated within the cluster borders (with a limit number of hops) to expand the association boundaries among sensor nodes based on their relationships shall also be considered to study their effectiveness in massive deployments.

**Author Contributions:** Conceptualization, G.M. and J.P.; Data curation, G.M.; Formal analysis, G.M. and J.P.; Funding acquisition, J.P.; Investigation, G.M. and J.P.; Methodology, G.M. and J.P.; Project administration, J.P.; Software, G.M.; Validation, G.M.; Writing—original draft, G.M.; Writing—review & editing, G.M. and J.P.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CH	Cluster Head
ETX	Expected Transmission Count
ID	Identifier
IN	Isolated Node
IoT	Internet of Things
LN	Linker Node
LQI	Link Quality Indicator
MDS	Mobile code Dissemination Strategy
MIA	Mobile Intelligent Agent
PoI	Point of Integration
RSSI	Received Signal Strength Indicator
SCm	Sensor Composition
SN	participant Sensor Node
USEQ	Update Sequence Number
WSN	Wireless Sensor Networks

## References

1. Mohanty, S.P.; Choppali, U.; Kougianos, E. Everything you wanted to know about smart cities: The Internet of things is the backbone. *IEEE Consum. Electron. Mag.* **2016**, *5*, 60–70. doi:10.1109/MCE.2016.2556879. [\[CrossRef\]](#)
2. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. doi:10.1016/J.ENG.2017.05.015. [\[CrossRef\]](#)
3. Perera, C.; Zaslavsky, A.B.; Christen, P.; Georgakopoulos, D. Sensing as a Service Model for Smart Cities Supported by Internet of Things. *CoRR* **2013**, arXiv:1307.8198.
4. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Things J.* **2016**, *3*, 854–864. doi:10.1109/JIOT.2016.2584538. [\[CrossRef\]](#)
5. Portilla, J.; Mujica, G.; Lee, J.; Riesgo, T. The Extreme Edge at the Bottom of the Internet of Things: A Review. *IEEE Sens. J.* **2019**, doi:10.1109/JSEN.2019.2891911. [\[CrossRef\]](#)
6. Kamalinejad, P.; Mahapatra, C.; Sheng, Z.; Mirabbasi, S.; Leung, V.C.M.; Guan, Y.L. Wireless energy harvesting for the Internet of Things. *IEEE Commun. Mag.* **2015**, *53*, 102–108. doi:10.1109/MCOM.2015.7120024. [\[CrossRef\]](#)
7. Basagni, S.; Naderi, M.Y.; Petrioli, C.; Spenza, D. Wireless Sensor Networks with Energy Harvesting. In *Mobile Ad Hoc Networking*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2013; Chapter 20, pp. 701–736. doi:10.1002/9781118511305.ch20.
8. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. doi:10.1109/JIOT.2016.2579198. [\[CrossRef\]](#)
9. Zorzi, M.; Gluhak, A.; Lange, S.; Bassi, A. From today's INTRANet of things to a future INTERNet of things: A wireless- and mobility-related view. *IEEE Wirel. Commun.* **2010**, *17*, 44–51. doi:10.1109/MWC.2010.5675777. [\[CrossRef\]](#)
10. Miorandi, D.; Sicari, S.; Pellegrini, F.D.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. doi:10.1016/j.adhoc.2012.02.016. [\[CrossRef\]](#)
11. Oteafy, S.; Hassanein, H.S. Resource re-use in wireless sensor networks: Realizing a synergetic internet of things. *J. Commun.* **2012**, *7*, 484–493. [\[CrossRef\]](#)
12. Levis, P.; Culler, D. *Maté: A Tiny Virtual Machine for Sensor Networks*; ACM Sigplan Notices; ACM: New York, NY, USA, 2002; Volume 37, pp. 85–95.
13. Dunkels, A.; Gronvall, B.; Voigt, T. Contiki—A lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, Tampa, FL, USA, 16–18 November 2004; pp. 455–462. doi:10.1109/LCN.2004.38. [\[CrossRef\]](#)

14. Dunkels, A.; Finne, N.; Eriksson, J.; Voigt, T. Run-time Dynamic Linking for Reprogramming Wireless Sensor Networks. In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys '06, Boulder, CO, USA, 31 October–3 November 2006; ACM: New York, NY, USA, 2006; pp. 15–28. doi:10.1145/1182807.1182810. [\[CrossRef\]](#)
15. Koshy, J.; Pandey, R. Remote incremental linking for energy-efficient reprogramming of sensor networks. In Proceedings of the Second European Workshop on Wireless Sensor Networks, Istanbul, Turkey, 2 February 2005; pp. 354–365. doi:10.1109/EWSN.2005.1462027. [\[CrossRef\]](#)
16. Tompkins, R.; Jones, T.B.; Nertney, R.E.; Smith, C.E.; Gilfeather-Crowley, P. Reconfiguration and management in Wireless Sensor Networks. In Proceedings of the 2011 IEEE Sensors Applications Symposium, San Antonio, TX, USA, 22–24 February 2011; pp. 39–44. doi:10.1109/SAS.2011.5739769. [\[CrossRef\]](#)
17. Hui, J.W.; Culler, D. The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04, Baltimore, MD, USA, 3–5 November 2004; ACM: New York, NY, USA, 2004; pp. 81–94. doi:10.1145/1031495.1031506. [\[CrossRef\]](#)
18. Panta, R.K.; Khalil, I.; Bagchi, S. Stream: Low Overhead Wireless Reprogramming for Sensor Networks. In Proceedings of the IEEE INFOCOM 2007—26th IEEE International Conference on Computer Communications, Barcelona, Spain, 6–12 May 2007; pp. 928–936. doi:10.1109/INFOCOM.2007.113. [\[CrossRef\]](#)
19. Zhong, X.; Navarro, M.; Villalba, G.; Liang, X.; Liang, Y. MobileDeluge: A Novel Mobile Code Dissemination Tool for WSNs. In Proceedings of the 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Philadelphia, PA, USA, 28–30 October 2014; pp. 537–538. doi:10.1109/MASS.2014.57. [\[CrossRef\]](#)
20. Marchiori, A.; Han, Q. PIM-WSN: Efficient multicast for IPv6 wireless sensor networks. In Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Lucca, Italy, 20–24 June 2011; pp. 1–6. doi:10.1109/WoWMoM.2011.5986469. [\[CrossRef\]](#)
21. Kim, D.; Nam, H.; Kim, D. Adaptive Code Dissemination Based on Link Quality in Wireless Sensor Networks. *IEEE Internet Things J.* **2017**, *4*, 685–695. doi:10.1109/JIOT.2016.2643659. [\[CrossRef\]](#)
22. He, D.; Chen, C.; Chan, S.; Bu, J. DiCode: DoS-Resistant and Distributed Code Dissemination in Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 1946–1956. doi:10.1109/TWC.2012.030812.111857. [\[CrossRef\]](#)
23. He, D.; Chan, S.; Guizani, M.; Yang, H.; Zhou, B. Secure and Distributed Data Discovery and Dissemination in Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1129–1139. doi:10.1109/TPDS.2014.2316830. [\[CrossRef\]](#)
24. Dong, C.; Yu, F. An efficient network reprogramming protocol for wireless sensor networks. *Comput. Commun.* **2015**, *55*, 41–50. doi:10.1016/j.comcom.2014.08.017. [\[CrossRef\]](#)
25. Mujica, G.; Rosello, V.; Portilla, J.; Riesgo, T. On-the-fly dynamic reprogramming mechanism for increasing the energy efficiency and supporting multi-experimental capabilities in WSNs. In Proceedings of the IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10–13 November 2013; pp. 5455–5460. doi:10.1109/IECON.2013.6700024. [\[CrossRef\]](#)
26. Guo, S.; He, L.; Gu, Y.; Jiang, B.; He, T. Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links. *IEEE Trans. Comput.* **2014**, *63*, 2787–2802. doi:10.1109/TC.2013.142. [\[CrossRef\]](#)
27. Zornoza, J.; Mujica, G.; Portilla, J.; Riesgo, T. Merging smart wearable devices and wireless mesh networks for collaborative sensing. In Proceedings of the 2017 32nd Conference on Design of Circuits and Integrated Systems (DCIS), Barcelona, Spain, 22–24 November 2017; pp. 1–6. doi:10.1109/DCIS.2017.8311637. [\[CrossRef\]](#)
28. Mujica, G.; Rosello, V.; Portilla, J.; Riesgo, T. Hardware-software integration platform for a WSN testbed based on cookies nodes. In Proceedings of the IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 25–28 October 2012; pp. 6013–6018. doi:10.1109/IECON.2012.6389099. [\[CrossRef\]](#)
29. Hexiwear. IoT and Wearables Development Platform. Available online: <http://www.hexiwear.com/> (accessed on 26 January 2017).
30. NXP. Kinetis KW40Z-2.4 GHz Dual Mode: BLE and 802.15.4 Wireless Radio Microcontroller (MCU) based on ARM Cortex-M0+ Core. Available online: <http://www.nxp.com/products/> (accessed on 14 March 2017).
31. TI. Second Generation 2.4 GHz ZigBee/IEEE 802.15.4 RF Transceiver. Available online: <http://www.ti.com/product/CC2520> (accessed on 23 June 2017).



32. De Couto, D.S.J.; Aguayo, D.; Bicket, J.; Morris, R. A High-throughput Path Metric for Multi-hop Wireless Routing. *Wirel. Netw.* **2005**, *11*, 419–434. doi:10.1007/s11276-005-1766-z. [CrossRef]
33. Xu, J.; Liu, W.; Lang, F.; Zhang, Y.; Wang, C. Distance measurement model based on RSSI in WSN. *Wirel. Sens. Netw.* **2010**, *2*, 606. [CrossRef]
34. Analog Devices. Single-Cycle 20 MIPS 8052 Core 8-bit Microcontroller. Available online: <http://www.analog.com/en/products/> (accessed on 7 May 2016).
35. FreeRTOS. Open-Source Real Time Operating System for Microcontrollers and Small Microprocessors. Available online: <http://www.freertos.org/> (accessed on 20 March 2017).
36. Experiment Setup Parameters and Main Results Data Considering the Comparison Of the Different Dissemination Strategies. Available online: <https://doi.org/10.5281/zenodo.2574159> (accessed on 20 February 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).