Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 328747, 14 pages
http://dx.doi.org/10.1155/2013/328747

*Research Article*

# Implementing a Distributed WSN Based on IPv6 for Ambient Monitoring

**D. F. Larios, J. M. Mora-Merchan, E. Personal, J. Barbancho, and C. León**

*Department of Electronic Technology, University of Seville, C. Virgen de Africa 7, 41011 Seville, Spain*

Correspondence should be addressed to D. F. Larios; dflarios@dte.us.es

Received 21 December 2012; Accepted 19 April 2013

Academic Editor: Adel Soudani



Traditionally, Wireless Sensor Networks (WSNs) are used for monitoring an extensive area. In these networks, a centralized server is usually used to collect and store the sensor information. However, new distributed protocols allow connections directly to the WSN nodes without the need of a centralized server. Moreover, these systems are able to establish communications among heterogeneous networks. The new protocols strategy is focused on considering several WSNs as a unique distributed one. This way, a user of the system is able to analyze a process under study as a whole instead of considering it as a set of different subsystems. This is the case in the evaluation of migratory waterbirds' environment. In this case, it is usual to deploy several WSNs in different breeding areas. They are all interconnected and they measure different environmental parameters. However, this improvement in the data access flexibility may result in a loss of network performance and an increase in network power consumption. Focused on this problem, this paper evaluates different communication protocols: distributed and centralized, in order to determine the best trade-off for environmental monitoring in different migratory areas of waterbirds.

## 1. Introduction

Nowadays, the number of electronic devices present in our environment is increasing continuously. The technological improvements and cost reduction of these smart devices provide a significant increase in their capabilities, among which are their connectivity. The *Internet of Things* (IoT) [1] is a paradigm which is gaining importance in the current scenario of modern telecommunication. This concept represents a novel scenario in which we are completely surrounded by smart devices. These devices can interact among them, providing different information or adding capabilities to the network [2]. An important group of these devices is formed by the Wireless Sensor Networks (WSNs) [3].

A WSN consists of many small devices deployed in a physical environment [4]. Each device, called a node, has special capabilities such as communication with its neighbors, sensing, and data storage and processing [5]. All nodes make a mesh network of devices that can collaborate among them allowing the implementation of distributed solutions to solve complex problems. Due to this, WSNs have many applications [6], among which environmental monitoring as an area where the potential impact is huge [7]. It allows monitoring an area at a low cost and little need of human presence. However, they have some technical requirements, such as the following.

(i) *Autonomy*. Batteries must be able to power the nodes during the whole network lifetime. Despite typical WSNs using low power radio devices (such as IEEE 802.15.4 [8] radio transceivers), the radio transceiver spends most of the energy consumption in a node in typical monitoring applications. Due to it, the network has to reduce data traffic as much as possible.

(ii) *Robustness*. In this kind of application, human maintenance is usually difficult because of the hardness of the terrain. Therefore, it is important to design robust networks that are adaptable to any incident.

(iii) *Flexibility*. The network must to be able to add, move, or remove nodes to meet the application requirements. The network must automatically detect the changes, organizing the communications in consequence.

(iv) *Low Price*. To save energy, the transmission range is limited too. This decrease in communication coverage is solved by increasing the density of the network. A high cost of the nodes would make unfeasible the use of a WSN versus other technologies.

WSNs are normally used in environmental monitoring applications to collect information through the sensors incorporated into each node. There is a special device called "Base Station" whose mission is to request and store the network information [9]. The Base Station generally acts also as a gateway, allowing the user to access the collected data through an infrastructured network, such as Internet.

To monitor an environment, the nodes of a WSN construct mesh networks that allow communications between devices. Due to the special characteristics of WSNs, such as their reduced energy available or low bandwidth, WSNs require the use of adapted protocols to establish communications. Nowadays, routing protocols with low power consumption continue being a main issue for WSNs. Several routing solutions have been proposed in the literature [10], mainly focused on energy consumption [11, 12], security issues [13, 14], or fault-tolerant capacity [15].

However, these proposed communication algorithms are classically designed to solve communication problems in an ad hoc manner. They solve particular scenarios, but they do not provide a general framework that allows intercommunication between heterogeneous networks with different communication requisites.

As an alternative to this classical WSN scheme and directly related to the IoT philosophy, some authors are currently proposing the use of IPv6 implementation over WSNs [16, 17]. This implementation are focused on reducing as much as possible the requirements of the Base Station, that is, using the Base Station only as a gateway between the two networks [18] and without intelligence. The use of IPv6 provides a common framework where additional protocols can be added, maintaining a basic interoperability between networks. IPv6 framework for low power consumption systems (also known as 6LoWPAN) is currently a main research area. Some authors are proposing new uses of 6LoWPAN technology [19, 20]. Other authors proposed additional schemes, for example, to compact addressing between devices [21] or to increase security [22, 23]. Other authors, as classical architectures, focused on provided efficient routing protocols [24, 25].

As can be seen, 6LoWPAN adds the IPv6 advantages of robustness and flexibility to WSNs, increasing the connectivity of the nodes. This implementation not only solves the retrieving information problem from the Base Station but it allows the access to every node in the WSN from anywhere in Internet [26]. Thanks to 6LoWPAN each node can be uniquely identified [27]. However, because IPv6 was not initially designed to operate over WSNs, it also has some constraints [28]. It could not make it interesting for all applications.

In this paper, we propose a comparison between IPv6 protocol implementation versus classical WSN communication protocols. This evaluation is focused on its use for monitoring application.

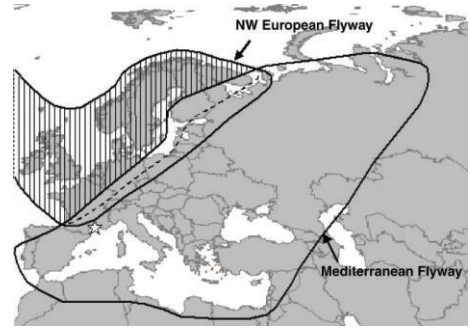



Figure 1: New potential delineations of Teal populations in western Eurasia [30].

The rest of the paper is organized as follows. Section 2 describes the application of our interest. A description and some common WSN protocols are described in Section 3. These protocols are evaluated in Section 4. Finally, Section 5 introduces a comparison between the evaluated protocols, before describing the final conclusions and remarks in Section 6.

## 2. Application Description

The evaluation of flooding areas, especially the estimation of flood level, is very important in monitoring waterbird colonies. Some studies relate the status of waterbird habits to the population of these kinds of birds [29]. Moreover, a change in the behavior and pathways of migratory waterbirds has been observed by biologists in the last decades [30], as can be seen in Figure 1. Due to this, biologists are very interested in this kind of environmental monitoring.

To capture that information, we already have a WSN deployed into the Doñana Biological Station [31]. Doñana is a wildlife reserve protected by the Spanish Government (located in the south of Spain). It covers a huge area of about 542 $km^2$ with little human interference through its entire history. This network is called ICARO [32] ("Inteligencia Computacional Aplicada a Redes de Observación," a Spanish acronym which means computational intelligence applied to monitoring networks). ICARO has been built based on TelosB [33] nodes and TinyOS [34] operating system. Each node (Figure 2) has a meteorological station with the ability to measure temperature, rainfall, wind speed and direction, and humidity. These nodes use a TelosB platform for processing and transmissions. Finally each node gets energy from a solar power system. These nodes use available information to predict the flood level of the marsh areas of Doñana [35].

ICARO network architecture is a centralized one, where the flood level estimation of each node is sent to a Base Station and stored later in a server. Node information is only accessible offline, and data is stored in a external database.

However, an effective monitoring of the habitat of migratory species requires deploying several networks in each

Figure 2: A node of ICARO WSN.

breeding area. Therefore, we are designing new WSNs (i.e., ICARO2, ICARO3, etc.) to cover new areas (the new design of the improved nodes is shown in Figure 3). One of our goals is to provide a unified interface for all the ICARO networks. This set of networks is called eSapiens, a Spanish acronym that means intelligent acquisition and processing system integrated in natural environments.

Although we can keep the ICARO architecture (for reusing the hardware and the deployments), this scheme suffers several problems: ICARO depends on several hard-defined bottle necks. All information goes through a sole Base Station. Inside nodes are not accessible from outside nodes. Therefore, it is not possible to access data directly from sources, but it is in a data server.

A new approach is chosen for eSapiens: flatten all hierarchical structures, allow direct communication to every node (Figure 4) in all networks, and keep historical data in each node. To embrace this approach, a change to a more complete communication protocol is necessary.

One advantage obtained from a more complete communication protocol is that it makes easy the communication between nodes of different networks, redirecting the information through a heterogeneous networks. Therefore, a node in an ICARO network can exchange information with another node of another ICARO network. It is depicted in Figure 5. Additionally, the information stored in a node can be retrieved by a PC using a standard web-page browser, where a user can access individually and transparently each device of the different ICARO networks into eSapiens infrastructure.

## 3. WSN Communications in Environmental Monitoring

Typically, WSN protocols allow mesh typologies of nodes with multihop structures. Thanks to this, it is possible to monitor huge areas with low cost and low power devices.

Figure 3: Node example for the new ICARO networks.

Moreover, it requires the use of low power radio transceivers. Typically, these radio transceivers are based on the IEEE 802.15.4 Standard [8]. This standard defines the physical and the MAC layers as is detailed in Figure 6. Based on this standard, several protocols have been defined. These protocols are necessary to enable the communication between nodes with multiple hops. Each node is required to know its neighbors and what routes it can use to send a message to another node, without sight line. Due to this, WSN protocols used for environmental monitoring require two classes of messages.

(i) *Network Messages*. These messages are used to maintain the network. It is used to discover node neighbors and obtain the routing table.

(ii) *Information Messages*. These messages are used to transmit useful information for application purposes. In environmental monitoring, these messages typically contain sensor measurements.

Both types of messages are needed; however the use of network messages increases the power consumption and increases the occupation index of the channel with information not directly related to the proposed application. Moreover, obtaining a high reliability requires the use of big headers. Therefore, to reduce the power consumption it is necessary to obtain a trade-off between the required reliability of the network and the amount of additional information sent between nodes.

Currently, there are different solutions for routing protocols in WSNs. Communication protocols designed specifically for WSNs, such as Collection Tree Protocol or ZigBee, are less flexible but their implementation over WSNs has less power consumption. On the other hand, IPv6 implementation, such as BLIP, takes advantage of the information messages to update the routing table.

In this section, we are going to describe some common WSN protocols, highlighting their advantages and disadvantages to use on the proposed application.

*3.1. ZigBee.* The ZigBee standard [36] is a project supported by the ZigBee Alliance. ZigBee defines the network layer and the application layer, both on top of IEEE 802.15.4 link layer. This standard defines several net structures in which three

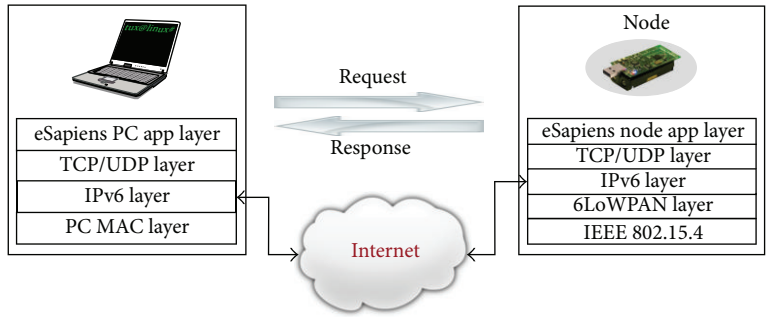



Figure 4: Direct communication to the node through Internet.

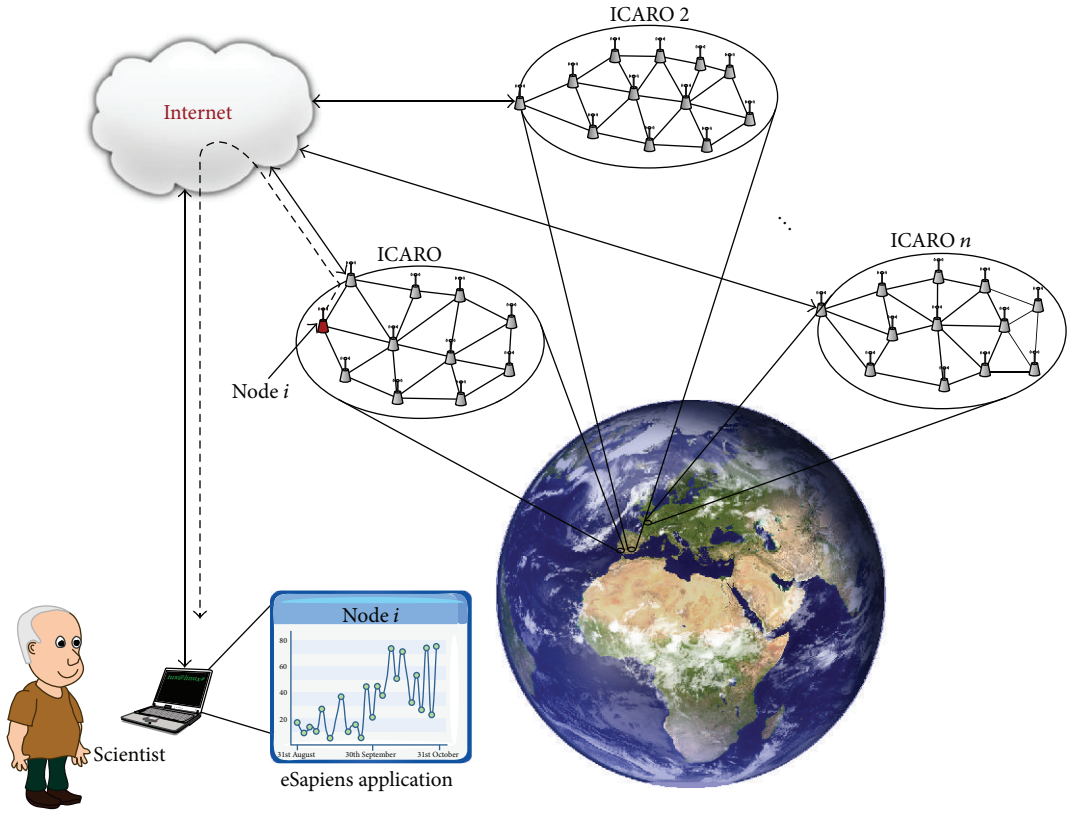



Figure 5: WSNs mutipoint access through Internet.

types of nodes are identified: *a coordinator*, *routers*, and *end devices*. The application layer ZigBee defines an application support sublayer (APS) on which the denominated end points (similar to TCP/IP ports) are defined. An application or object can be defined for each end point. However, there is a reserved end point for the ZigBee device object (ZDO) which is responsible for communicating the *binding tables* to the network. These tables keep information about the network nodes' presence and their services. In the ZigBee network the services define the *profiles* whereby the devices are grouped in *clusters* (devices within the same profile).

*3.2. Collection Tree Protocol.* The Collection Tree Protocol (CTP) [37] is a tree-based collection protocol. In this protocol, a single or small group is shown as sink nodes (tree roots) which are the main branches of the tree (connectivity between nodes). Based on this tree, when each node sends a message to a sink node, it looks for the most suitable neighbor node, using a routing gradient. With this technique, the protocol is responsible for network management discovering the neighbors and estimating the best transmission path (minimizing the number of hops and the global consumption). This is possible because each node has a table with a list of the best neighbors to reach a root node. The CTP is common in data collection applications where nodes periodically send information to a root node (typically called Base Station). A specific implementation of this protocol is CTP Noe [38].

*3.3. 6LoWPAN.* The IPv6 in Low Power Wireless Personal Area Networks (6LoWPAN) [39, 40] is a project supported

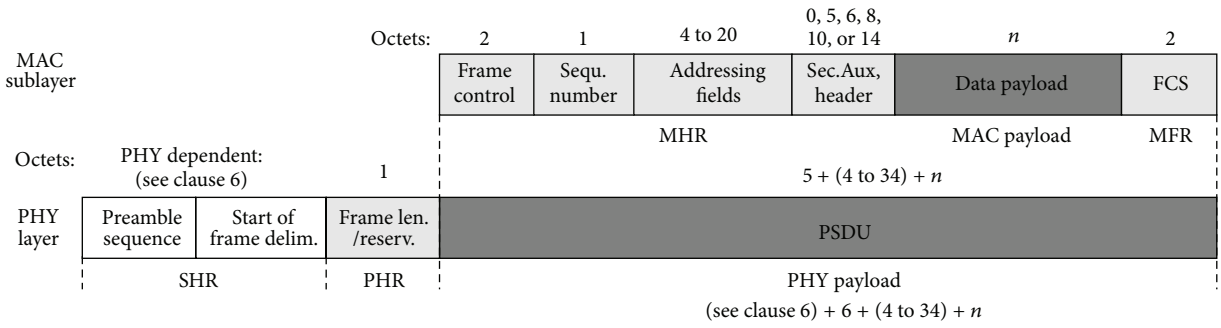



Figure 6: IEEE 802.15.4 datagram.

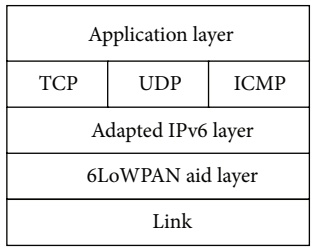



Figure 7: 6LoWPAN layer description.

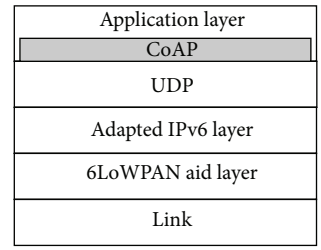



Figure 8: CoAP layer description.

by the IETF (Internet Engineering Task Force). It was created to adapt IPv6 datagrams over IEEE 802.15.4 links. This adaptation presents the problem of great differences in MTU sizes between IPv6 and IEEE 802.15.4 (much larger in typical IPv6 implementation). To resolve this problem, 6LoWPAN implements a packet fragmentation mechanism [41] and a header compression algorithm [42]. On the other hand, the use of IPv6 simplifies tasks of network management and expands connectivity with Internet protocols (e.g., TCP, UDP, and ICMP) and services [26] (e.g., HTTP, SMTP, FTP, and SOAP), as can be seen in Figure 7.

*3.3.1. BLIP.* The Berkeley Low power IP stack (BLIP) [43] is implementation of IPv6 for WSNs over TinyOS. Currently, it is not full IETF standard compliant. However, it supports 6LoWPAN/HC-01, header compression, IPv6 neighbor discovery, default route selection, point-to-point routing, and network programming. If there are communications in the network, the device that receives a message analyzes the sender information and the ACK to refresh its neighbor and the routes tables. Only in the case that no communication occurs during a long time, a node sends a message to refresh its routing table. Additionally, some of the implementation allow the use of compressed headers. That reduces the overhead of IPv6 communications. With all this, BLIP provides significant interoperability with other IP networks.

*3.3.2. Constrained Application Protocol.* The Constrained Application Protocol (CoAP) [44] is a specialized web transfer protocol for machine-to-machine (M2 M) applications. It is proposed by CoRE (Constrained RESTful Environments), a work group at the IETF. The main goal of CoAP is to provide a mechanism to easily translate a protocol like HTTP to a less complex one. It allows integration between constrained networks (such as WSNs) and standard Internet networks. CoAP is a single protocol over UDP (as is shown in Figure 8) and is subdivided into two sublayers. The first one, the CoAP message sublayer, is responsible for dealing with UDP (CoAP operates over UDP) and defines four types of messages: *Confirmable*, *Nonconfirmable*, *Acknowledgement*, and *Reset*. The second one is the request/response sublayer. It contains methods and response codes. An implementation of this protocol is `libcoap` which provides the same methods as the ones used by HTTP: GET, PUT, POST, and DELETE. However, there is specific implementation of these libraries for WSNs [45] with a more reduced set (only GET and PUT) which is typically sufficient for most applications.

## 4. Experimental Evaluations

In this paper, we are going to compare different protocols which allow us to use IPv6 in a WSN (a network of TelosB nodes with TinyOS Operating System). Additionally, some other well-known protocols are included in the comparison.

A developed test application sends the same application data using each studied protocol. A sniffer captured the whole wireless traffic between nodes. To acquire data traffic, our testbench comes with a USB dongle 802.15.4 sniffer (IA OEM DAUB1 2400 by Adaptive Modules Ltd.) [46] and "Perytons Protocol Analyzers" [47] (Figure 9) software to analyze the packets detected.

The network used to test the protocols is made of 3 nodes with the architecture described in Figure 10.

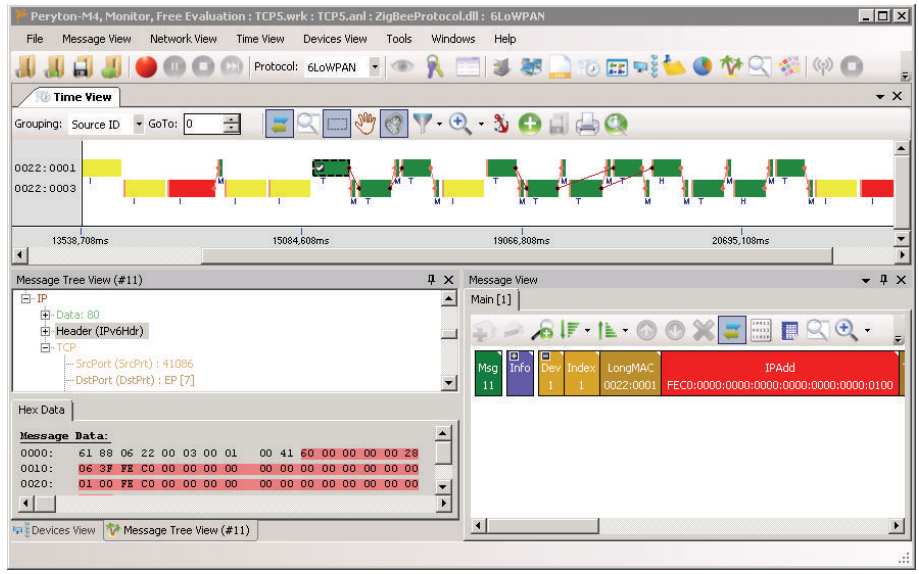
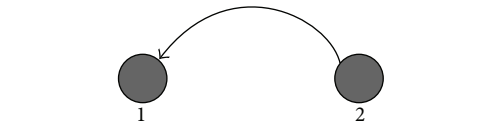
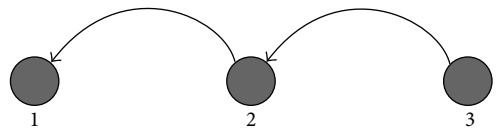



Figure 9: Perytons Protocol Analyzers screenshot.




(a) Topology without hop




(b) Topology with hop

Figure 10: Studied topologies.

A limitation of output power in the transceiver of nodes (down to −12 dB) allows the experimentation in a close controlled scenario. It causes each node not to have full connectivity with the rest of the nodes. Due to this, in some cases nodes have to route messages through other nodes. Same coverage is used for all the experiments.

All the IPv6 dependent protocols use the Berkeley implementation for WSNs (6LoWPAN).

Our test application is developed to imitate the behavior of the proposed flooding estimation application.

(i) Every node is cyclically acquiring environmental information.

(ii) Once per day, a WSN node executes the data aggregation algorithm, to predict flood level.

(iii) Only if the flood level is different from the last estimation, it sends a message to the Base Station with the new estimation.

Based on this simple architecture, a count of sent bytes per data has been done. In all measurements, it only showed size in bytes of any layer above MAC layer. To get real size of messages, it is necessary to add the MAC size. Usually WSNs based on IEE802.15.4 radio transceiver use short addresses (9 bytes) to reduce the overhead. When some protocols depend on responses in MAC layer (i.e., wait for MAC ACK), those messages are shown too.

Moreover, all experiments done in both topologies show an $n$-hop communication which is an equivalent to $n$ times 1-hop. The overhead of frames for the evaluated protocols does not increase with multiple hops. So, the results only show the last transmission (1-hop).

All the expressed results are obtained without considering fragmentation. To maintain a low power consumption, it is necessary to reduce as much fragmented messages as possible because it increases the required transmissions with their associated ACK and therefore it increases the overhead.

*4.1. Static Ad Hoc Implementation.* This implementation is used only for comparison purposes. An Ad hoc implementation is specific to application and network. The efficiency of an ad hoc implementation is maximum because its topology is fixed in programming time and routing tables are static. Due to this, these networks do not require network messages to obtain the topology neither to provide information to help the routing of the packets.

Against that, it needs to know beforehand the location of the network nodes. A change in its topology requires reprogramming all nodes. Neither of these problems have been considered here.

This kind of implementation is not very common due to the difficulty in the evaluation of topologies in networks.

Despite their lack of flexibility, these implementation have very few overheads and do not require additional messages to build a routing table. So, it is considered as an ideal case and it is used as reference to compare other routing protocols.

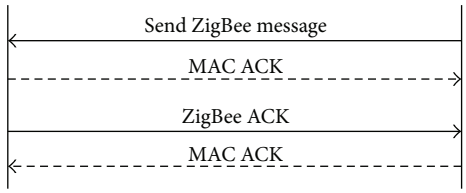



Figure 11: ZigBee message transmission.

With the proposed testbench, we talk about two different protocols: a one *similar* to UDP without ACK (1 byte to destination, 1 byte to source and payload) and a one *similar* to TCP with ACK (2 bytes more with source and target to send ACK) (see Table 1). If both protocols do not coexist, it is not necessary to have an extra byte to identify the protocol.

This implementation does not allow communication between devices. Moreover, they cannot acquire information from a remote WSN. Therefore, all the acquired information must be stored in a central server.

*4.2. ZigBee.* The structure of ZigBee transmissions is depicted in Figure 11. In this case, a remote node is sending a non-requested message. ZigBee requires the use of ACK messages at link and MAC layers. Figure 12 shows a ZigBee frame. As can be seen, it has a reduced overhead, similar to CTP one.

Table 2 sums up the size of messages transmitted with ZigBee. These messages do not increase in size with multihop transmissions.

ZigBee allows direct communication between devices of a network, but it does not allow to redirect messages with a remote network. ZigBee has not been designed to allow fragmentation.

*4.3. Collection Tree Protocol Implementation.* Collection protocols are useful to gather information from different sensors to a central device in a network, generally to the Base Station. This is the typical application in environmental monitoring where all the information gathered in the Base Station is stored in a database.

Collection nodes only store a reduced number of routes. They try to send information through an optimum path to a Base Station. If the nodes in the path are busy, they search for other paths, always trying to minimize the cost (estimated as number of hops).

This implementation does not allow communication between devices. The network only maintains routes to the Base Station. Moreover, it cannot acquire information from other remote WSNs. Therefore, all the acquired information must be stored in a central server.

Figure 13 shows a generic frame of a collection message. Its overhead is reduced.

Typical CTP schema is depicted in Figure 14. This protocol only sends ACK at MAC layer, maintaining the number of exchanged messages low.

Table 3 sums up the size of messages transmitted with the evaluated CTP protocol (from TinyOs Stack). CTP messages do not increase their size in multihop nodes.

Table 1: Ad hoc protocol.

| Msg type | Bytes sent |
|---|---|
| Data message | 2 + payload (up to 114) |
| MAC ACK | 5 |

Table 2: ZigBee protocol.

| Msg type | Bytes sent |
|---|---|
| ZigBee message | 18 + payload (up to 100) |
| ZigBee ACK | 18 |
| MAC ACK | 5 |

Table 3: Collection protocol.

| Msg type | Bytes sent |
|---|---|
| Data message | 12 + payload (up to 106) |
| MAC ACK | 5 |

*4.4. TCP BLIP Implementation.* BLIP implementation supports over the 6LoWPAN aid layer. The scheme of a TCP message using 6LoWPAN is depicted in Figure 15.

TCP transmission require a complex message negotiation. This negotiation ensures the integrity of the information, but drastically increases the overhead. Messages vary in function of the application. For example, the scheme to transmit a webpage over 6LoWPAN is depicted in Figure 16.

As can be seen, a webpage structure increases the number of required messages even more. Due to this, in transmission with constrained communications, such as that used in WSNs, it is better to use transmissions without a protocol in application layer, that is, sending the information after establishing the socket connection.

TCP allows communication between devices, whether they are in the same network or not. Moreover, the accessibility of each device in a TCP network allows the user to request information stored locally. So, transmission is only by demand.

TCP BLIP permits message fragmentation. If a message is higher than the maximum payload, BLIP automatically fragments it. To do this, it adds fragmentation header to the first fragment (4 bytes). This header is added between the MAC header and the 6LoWPAN header.

The rest of the fragments are sent only with MAC header and a 5-byte fragmentation header that identifies the full message. The rest of the headers in these fragments are avoided to reduce the overhead as much as possible.

*4.4.1. TCP BLIP without Header Compression.* Figure 17 depicts a standard payload message, using TCP frames without header compression. TCP frames require the use of big headers. It considerably reduces the size of the payload or fragments the message.

Table 4 sums up the size of messages with a TCP connection without header compression. As can be seen, the number of transmitted messages and the number of bytes sent in communications are high.

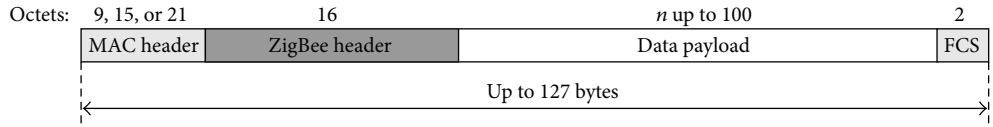



Figure 12: ZigBee frame.

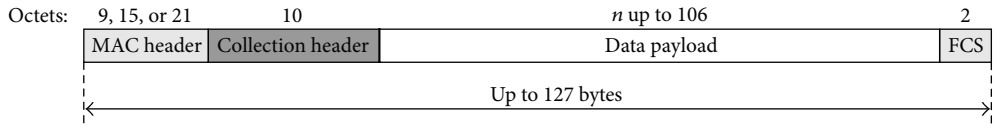



Figure 13: CTP frame.

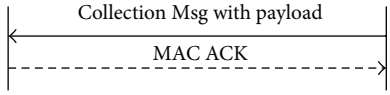



Figure 14: CTP message transmission.

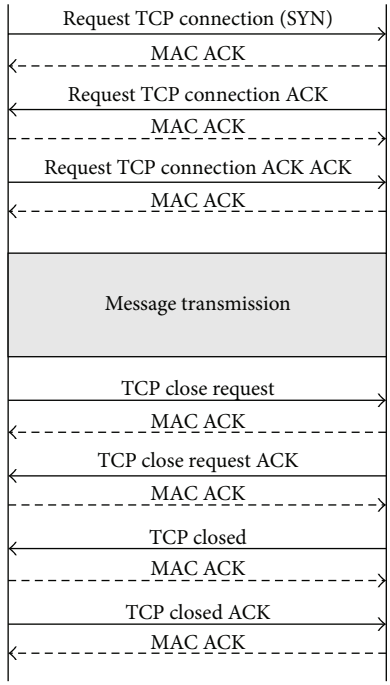



Figure 15: TCP negotiation.

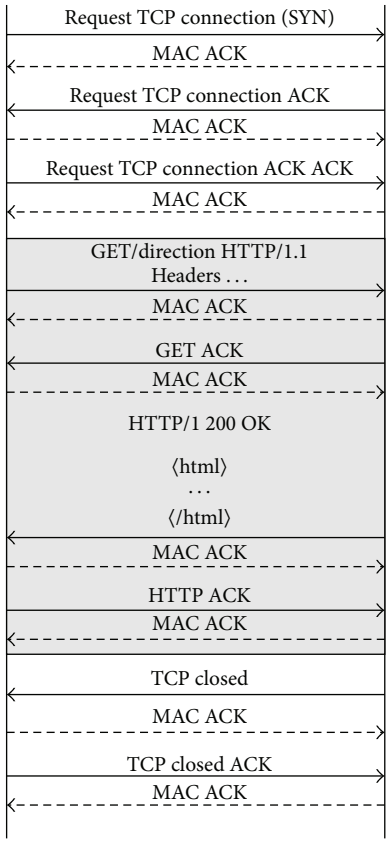



Figure 16: Webpage transmission.

TCP requires a high number of transmissions. Firstly, it requires to start explicit connection with an SYN message. After that, we need to send the message with the payload. Finally, it is necessary to close the connection explicitly. Moreover, TCP requires the use of the ACK messages link layer, in addition to the ACK in MAC layer. Due to this, the TCP protocol is in general too costly to be used in devices with limited bandwidth, such as WSN devices.

*4.4.2. TCP BLIP with Header Compression.* Figure 18 depicted a TCP payload frame. The size of messages with header compression is lower than that of messages without it. However, their headers are still too high for a protocol with constrained maximum message size, such as in 802.15.4.

Table 5 sums up the size of transmission messages in a TCP connection with header compression. As can be seen, TCP header compression reduces the overhead, but it does not reduce the number of transmissions.

As conclusion, the TCP over 802.15.4 radio transceiver requires a large number of transmissions and it has a big overhead, even with header compression. TCP is only useful with not so frequent communication with reduced payload,

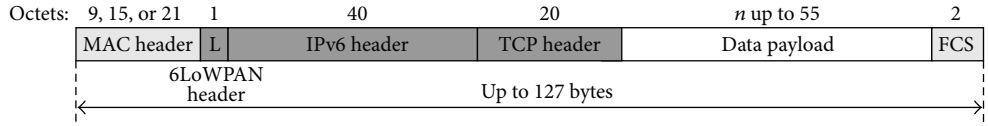



Figure 17: TCP frame without compression.

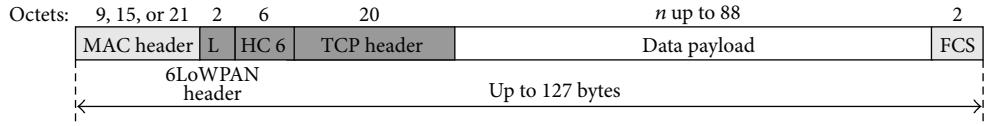



Figure 18: TCP frame with compression.

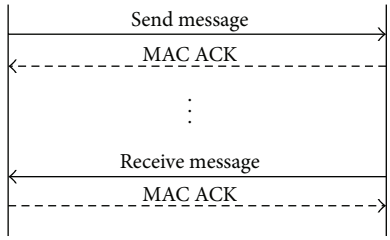



Figure 19: UDP transmission.

Table 4: TCP without compression.

| Msg type | Bytes sent |
|---|---|
| Request TCP connection (SYN) | 83 |
| Request TCP connection ACK | 71 |
| Request TCP connection ACK ACK | 63 |
| Data message | 63 + payload (up to 55) |
| ACK | 71 |
| MAC ACK | 5 |
| Close request | 73 |
| Close ACK | 63 |
| Closed | 63 |
| Closed ACK | 63 |

where the system reliability is more important than its power consumption.

*4.5. UDP BLIP Implementation.* In contrast to TCP connection, UDP does not have negotiation to ensure correct transmission. Due to it, UDP needs to send less messages. Therefore, it is not reliable (i.e., there is no guarantee that sent UDP messages or packets would reach their destinations at all). Figure 19 shows a scheme of an UDP connection.

Despite its less overhead, the lack of reliability of UDP messages is a drawback, especially in wireless communication with a nonnegligible packet error rate, such as 802.15.4 communications.

Due to this, to increase the reliability in WSNs using UDP, it is necessary to add a communication protocol at the application layer, such as CoAP or an adaptation [48] of IEEE1451 Standard [49].

Table 5: TCP with compression.

| Msg type | Bytes sent |
|---|---|
| Request TCP connection (SYN) | 50 |
| Request TCP connection ACK | 38 |
| Request TCP connection ACK ACK | 30 |
| Data message | 30 + payload (up to 88) |
| ACK | 38 |
| MAC ACK | 5 |
| Close request | 40 |
| Close ACK | 30 |
| Closed | 30 |
| Closed ACK | 30 |

Table 6: UDP without compression.

| Msg Type | Bytes sent |
|---|---|
| Data message | 51 + payload (up to 67) |
| MAC ACK | 5 |

Like in the TCP BLIP implementation, if messages are longer than the available payload, they are fragmented. This fragmentation has the same structure of TCP communications: first fragment adds a header fragmentation (4 bytes) to the original header. The rest of the fragments only have fragmentation header of 5 bytes without any other header.

As TCP protocol, UDP allows communication between devices, whether they are in the same network or not, and it also allows to store information locally, transmitting it only on demand to a user.

*4.5.1. UDP BLIP without Header Compression.* Figure 20 summarizes a UDP frame between two nodes, without multihop, without header compressions or security.

As can be seen, it has less overhead than TCP, but it does not ensure the receiving of the message.

Table 6 summarizes the number of bytes required to send a message between nodes with UDP without compression.

UDP without compression significantly reduces the overhead in comparison with TCP connection.

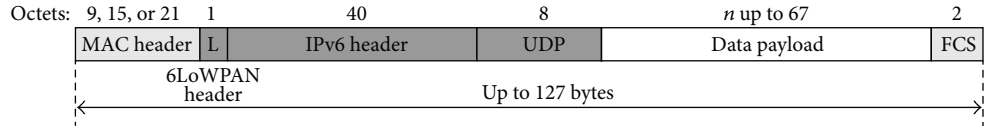



Figure 20: UDP frame without compression.

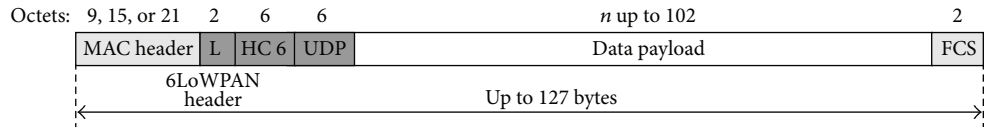



Figure 21: UDP frame with compression.

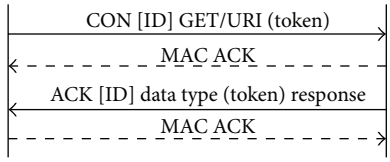



Figure 22: CoAP transmission.

*4.5.2. UDP BLIP with Header Compression.* Figure 21 summarizes a UDP frame between two nodes, without multihop, with header compression and security.

As can be seen, it has very low overhead, especially if short MAC addresses are used.

Table 7 summarizes the number of bytes needed to send a UDP message between nodes with compression.

The overhead of UDP communications is slightly higher than the case of CTP networks, but it allows more flexibility. As it was mentioned before, its main drawback is the lack of reliability.

*4.6. CoAP Implementation.* CoAP implementation add a minimal negotiation to UDP messages with the purpose of increasing the reliability, but maintaining the overhead low. Figure 22 shows the structure of a CoAP negotiation.

The use of Piggy-backed messages prevents the use of a link level ACK, reducing the traffic. Moreover, it uses an implicit socket connection. Due to this, it does not require additional messages to establish or close connections. Like TCP or UDP, CoAP is able to communicate between devices which allow transmittng data on demand.

*4.6.1. CoAP without Header Compression.* The overhead of most common CoAP messages are depicted in Table 8 and Figure 23.

This protocol is a good trade-off between reliability and overhead. It does not increase too much the overhead, but establishes messages to ensure a correct transmission.

*4.6.2. CoAP with Header Compression.* Figure 24 depicted a CoAP frame with header compression. As can be seen, this protocol has a reduced overhead.

Table 7: UDP with compression.

| Msg type | Bytes sent |
|---|---|
| Data message | 16 + payload (up to 102) |
| MAC ACK | 5 |

Table 8: CoAP w/o compression and w/o fragmentation.

| Msg type | Bytes sent |
|---|---|
| GET | 56 + token (0 to 4) + payload (up to 62) |
| POST | 56 + token (0 to 4) + payload (up to 62) |
| MAC ACK | 5 |

Table 9: CoAP w/ compression and w/o fragmentation.

| Msg type | Bytes sent |
|---|---|
| GET | 21 + token (0 to 4) + payload (up to 97) |
| POST | 21 + token (0 to 4) + payload (up to 97) |
| MAC ACK | 5 |

Table 9 sums up the results obtained with the most common CoAP messages used in 802.15.4 WSNs. It is important to consider that CoAP implementation for *TinyOS* is still a work in progress, and not all the methods are currently available.

The reliability and extra cost are similar to CTP protocols or ZigBee, but CoAP provides more flexibility.

In conclusion, CoAP is presented as an interesting compromise between the reliability of TCP and the reduced overhead of UDP. Its overhead is reduced, but nonetheless, it is higher in the case of CTP messages.

## 5. Comparison between Protocols

This section describes a comparison between the different tested protocols.

*5.1. Routing Overheads and Evaluation of Power Consumption.* Routed messages between networks require the use of headers. But headers increase the number of total sent bytes in

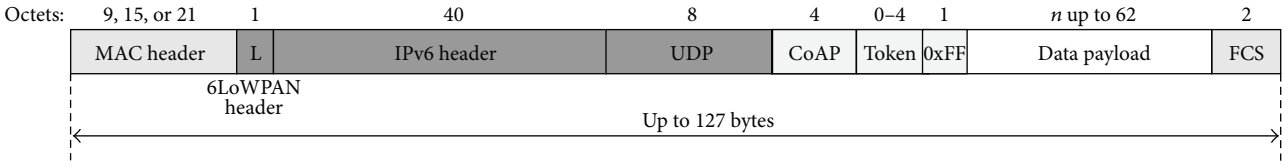



Figure 23: CoAP frame without compression.

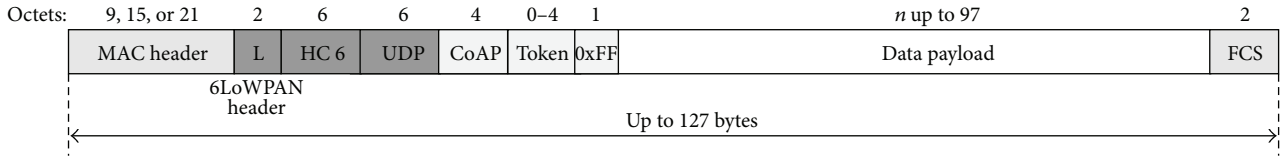



Figure 24: CoAP frame with compression.

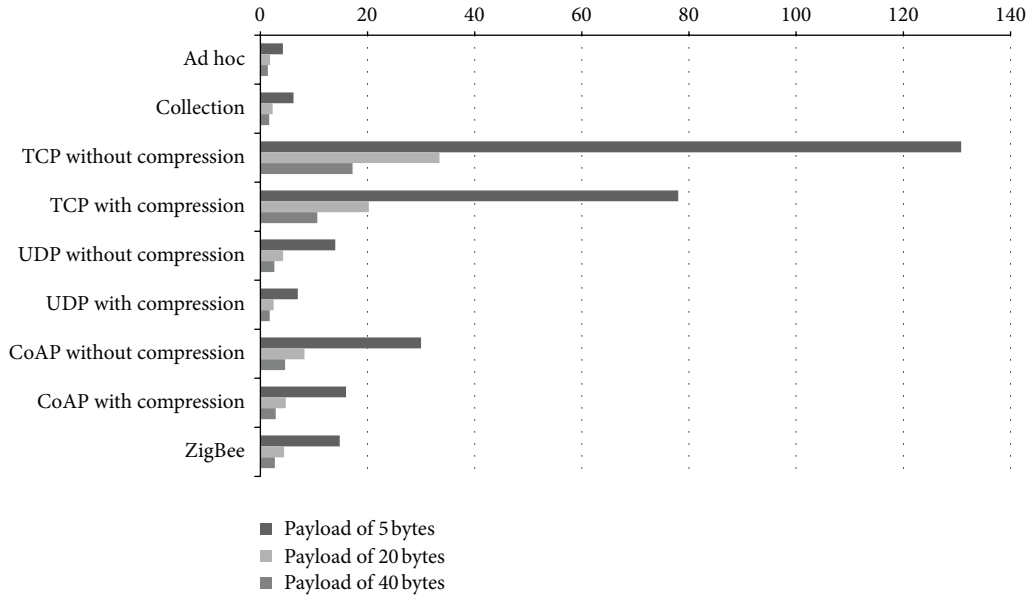



Figure 25: Overhead factor versus payload.

the network. Therefore, it is necessary to search for a trade-off between size of header and reliability. Table 10 shows the number of messages interchanged in order to send a packet of data (5 bytes) and total number of sent bytes (including our payload of 5 bytes).

The ratio between total number of bytes versus payload is the overhead factor. Overhead factor (also shown in Table 10) is related directly to energy consumption. The time of transmission or, in other words, the number of sent bytes are the main influence on energy consumption. Bigger overhead implies bigger energy consumption for the same payload, as energy consumption is the main issue in Wireless Sensor Networks [11] because of limited autonomy in nodes.

To evaluate the energy consumption of sending a packet, it can be estimated as described in [35]:

$$E_{Tx,i} = T_{Tx} \cdot P_{Tx}, \tag{1}$$

where $T_{Tx}$ is time of transmission and $P_{Tx}$ is power of emission. $P_{Tx}$ is estimated as 38 mW [33].

At the same time, there is some energy consumption in the receiving node. It can be estimated as

$$E_{Rx,j} = T_{\text{on}} \cdot P_{Rx}, \tag{2}$$

where $T_{\text{on}}$ is the time that its transceiver is active and $P_{Rx}$ is the power consumed in reception (41 mW [33]). $T_{\text{on}}$ depends on energy saving policies on each protocol.

So, if we only consider energy losses in transmission time, that energy can be modeled as $E_L$:

$$E_L = E_{Tx} + E_{Rx} = \frac{l_s}{r_t} \cdot \left(P_{Tx} + P_{Rx}\right), \tag{3}$$

where $r_t$ is the transmission rate of the platform (i.e., $r_t$ = 250 kb/s) and $l_s$ is the total amount of bits sent in communication process. Based on data of Table 10, Table 11 shows estimation of energy losses in transmission calculated on each protocol.

Table 10: Protocol comparison (payload 5 bytes without fragmentation).

| Protocol | Msg interchanged | Bytes sent | Overhead factor |
|---|---|---|---|
| Ad-hoc with ACK | 2 | 21 | 4.2 |
| Collection | 2 | 31 | 6.2 |
| TCP (w/o compression) | 16 | 654 | 130.8 |
| TCP (w/ compression) | 16 | 390 | 78.0 |
| UDP (w/o compression) | 2 | 70 | 14.0 |
| UDP (w/ compression) | 2 | 35 | 7.0 |
| CoAP (w/o compression) | 4 | 75 | 30.0 |
| CoAP (w/ compression) | 4 | 40 | 16.0 |
| ZigBee | 4 | 74 | 14.8 |

The overhead factor depends on the length of headers and payload. As payload increases, the overhead will decrease. Figure 25 shows the evolution of the overhead factor versus payload size for each tested protocol.

As can be seen, the use of an ad hoc solution has reduced headers, but this system provides almost no services. The use of complete TCP/IP headers only justified sending large messages. UDP and CoAP offer good ratios of overhead with a limited set of services.

To estimate total global consumption, it is necessary to add all the messages of network construction. This amount depends on protocol and topology and they are arranged based on the design time of the application.

*5.2. Latency.* Table 12 depicts the latency between the different evaluated protocols. Latency is obtained measuring the time between starting a request of new information and the time when the node receives this information.

TCP latency is several times higher than the rest of the evaluated protocols. UDP and CoAP have similar latency, with CTP having the lowest latency (approximately, 5 times lower than UDP). Despite of the extra information added by CoAP, it presents a similar latency to UDP.

*5.3. Evaluation of the Comparison.* According to the above results, the main advantages and disadvantages of the evaluated protocols are depicted in Table 13.

For a distributed application, such as the proposed flood level monitoring for waterbirds, 6LoWPAN based protocols are the best trade-off between flexibility and power consumption.

Among the evaluated protocols based on 6LoWPAN, CoAP is the best option for constrained networks. This protocol has advantages such as reliability, but it maintains a low overhead. Moreover, its last draft [50] provides techniques to reduce power consumption like using local proxies or sleeping radio transceiver.

Table 11: Energy consumption (estimated for transmission of 5-byte message without fragmentation).

| Protocol | Energy consumption/(Ws) |
|---|---|
| Ad hoc with ACK | $53 \cdot 10^{-6}$ |
| Collection | $78 \cdot 10^{-6}$ |
| TCP (w/o compression) | $1653 \cdot 10^{-3}$ |
| TCP (w/ compression) | $986 \cdot 10^{-6}$ |
| UDP (w/o compression) | $177 \cdot 10^{-6}$ |
| UDP (w/ compression) | $88 \cdot 10^{-6}$ |
| CoAP (w/o compression) | $190 \cdot 10^{-6}$ |
| CoAP (w/ compression) | $101 \cdot 10^{-6}$ |
| ZigBee | $187 \cdot 10^{-6}$ |

Table 12: Latency comparison between protocols.

| Protocol | Compressed/ms | Not compressed/ms |
|---|---|---|
| TCP | 497 | 395 |
| UDP | 18 | 25 |
| CoAP | 21 | 29 |
| ZigBee | — | 13 |
| CTP | — | 4 |

The use of local storage and direct communication between devices reduces interchanged messages to Base Station and they avoid the maintenance of a central server. The maximum storage of information depends on the memory of platform and the number of nodes. That is, TelosB nodes have an external Flash of 1 Mb, and for flood level estimation, we need 5 bytes in each change (4 bytes with a timestamp + a byte with the estimated flood level). In a worst case scenery with a daily flood level modification, every node can retain up to 7 months of information.

## 6. Conclusions

This paper proposes eSapiens, a distributed WSN for monitoring flood level in several breeding areas of migratory waterbirds. The used data aggregation algorithm allows the use of local storage. Thus it avoids the use of a central server, simplifies the architecture, and reduces the cost. Moreover, the proposed infrastructure requires communication between remote devices. This architecture offers a trade-off between power consumption and reliability.

Focusing on these issues some algorithms have been evaluated. These algorithms can be divided into two families: classic centralized algorithms and fully distributed algorithms.

According to our conclusions, current fully distributed algorithms, such as IPv6 over WSNs, provide flexibility without too much extra cost. For all this, eSapiens has chosen CoAP as best option for its IEEE 802.15.4 WSN devices.

Currently, the authors are developing additional WSNs to spread in other flooded areas where waterbirds live.

Table 13: Comparison of evaluated protocols.

| Protocol | Ad-hoc | ZigBee | CTP | 6LoWPAN (TCP) | 6LoWPAN (UDP) | 6LoWPAN (CoAP) |
|---|---|---|---|---|---|---|
| Mesh network (allows direct communication between nodes) | No | Yes | No | Yes | Yes | Yes |
| Redirection (allows communication between nodes of different networks) | No | No | No | Yes | Yes | Yes |
| Central node required to store gathered information | Yes | No | Yes | No | No | No |
| Communication node → Internet | No | No | No | Yes | Yes | Yes |
| Communication node ← Internet | No | No | No | Yes | Yes | Yes |
| Overhead | Very low | Low | Low | Very high | Low | Medium |
| Latency | Very low | Low | Very low | Very high | Low | Low |
| Energy consumption | Very low | Low | Very low | Very high | Medium | Medium |

## Acknowledgments

This research has been supported by the Consejería de Innovación, Ciencia y Empresa, Junta de Andalucía, Spain, through the projects of excellency ARTICA (Reference no. P07-TIC-02476) and eSapiens (Reference no. TIC-5705) and by the "Cátedra de Telefónica, Inteligencia en la Red," Seville, Spain, through the project TORTUGA.

## References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] A. Jara, M. Zamora, and A. Skarmeta, "Glowbal IP: an adaptive and transparent IPv6 integration in the internet of things," *Mobile Information Systems*, vol. 8, no. 3, pp. 177–197, 2012.

[3] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-Fi enabled sensors for internet of things: a practical approach," *IEEE Communications Magazine*, vol. 50, pp. 134–1143, 2012.

[4] M. Islam, M. Hassan, G.-W. Lee, and E. -N. Huh, "A survey on virtualization of wireless sensor networks," *Sensors*, vol. 12, no. 2, pp. 2175–2207, 2012.

[5] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[6] I. Akyildiz, Y. Su, W. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[7] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and I. Robla, "A Review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends," *Sensors*, vol. 9, no. 6, pp. 4728–4750, 2009.

[8] *IEEE Std 802.15.4-2003: IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LRWPANs)*, 2003.

[9] R. Machado, N. Ansari, G. Wang, and S. Tekinay, "Adaptive density control in heterogeneous wireless sensor networks with and without power management," *IET Communications*, vol. 4, no. 7, pp. 758–767, 2010.

[10] M. Haneef and D. Zhongliang, "Design challenges and comparative analysis of cluster based routing protocols used in wireless sensor networks for improving network life time," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 1, pp. 450–459, 2012.

[11] L. Liu and C. Wu, "A novel power intensity routing in WSN," *International Journal of Digital Content Technology and Its Applications*, vol. 6, no. 1, pp. 178–184, 2012.

[12] D. Ding, L. Fangai, L. Qianqian, and Y. Guangxu, "An improved clustering algorithm based on backup path," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 8, pp. 207–216, 2012.

[13] S. Zihao and L. Shufen, "Security threats and security policy in wireless sensor networks," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 10, pp. 166–173, 2012.

[14] C. Fenhua, "A distributed dynamic pairwise key establishment scheme for wireless sensor networks," *International Journal of Advancements in Computing Technology*, vol. 4, no. 4, pp. 261–267, 2012.

[15] H. Li, L. Pang, and Y. Wang, "A domain-based secure communication scheme with fault-tolerant capacity," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 5, pp. 44–52, 2012.

[16] J. Ko, A. Terzis, S. Dawson-Haggerty, D. Culler, J. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 96–101, 2011.

[17] S. Hong, D. Kim, M. Ha et al., "SNAIL: an IP-based wireless sensor network approach to the Internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010.

[18] M. Ha, S. Kim, H. Kim, K. Kwon, N. Giang, and D. Kim, "SNAIL gateway: dual mode wireless access points for WiFi and IP based wireless sensor networks in the internet of things," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '12)*, pp. 169–173, 2012.

[19] N.-T. Dinh and Y. Kim, "Restful architecture of wireless sensor network for building management system," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 1, pp. 46–63, 2012.

[20] A. Jara, D. Fernndez, P. Lpez, M. Zamora, L. Marin, and A. Skarmeta, "YOAPY: a data aggregation and pre-processing module for enabling continuous healthcare monitoring in

the internet of things," in *Proceedings of the 4th International Workshop on Ambient Assisted Living (IWAAL '12)*, vol. 7657, pp. 248–255, Lecture Notes in Computer Science, 2012.

[21] Y.-J. Wang, Z.-H. Qian, X. Wang, and D.-Y. Sun, "Addressing scheme for internet of things based on ipv6 over low-power wireless personal area network (6LoWPAN)," *Journal of Electronics and Information Technology*, vol. 34, no. 4, pp. 763–769, 2012.

[22] H. Yu and J. He, "Trust-based mutual authentication for bootstrapping in 6LoWPAN," *Journal of Communications*, vol. 7, no. SPL.ISS. 8, pp. 634–642, 2012.

[23] L. Oliveira, J. Rodrigues, A. De Sousa, and J. Lloret, "Denial of service mitigation approach for ipv6-enabled smart object networks," *Concurrency Computation Practice and Experience*, vol. 25, no. 1, pp. 129–142, 2013.

[24] R. Lu, X. Li, J. Wang, and F. Sun, "Research on route protocol and architecture for wsn based on ipv6," *Advanced Materials Research*, vol. 616–618, pp. 2233–2238, 2013.

[25] A. Castellani, M. Rossi, and M. Zorzi, "Back pressure congestion control for CoAP/6LoWPAN networks," *Ad Hoc Networks*, 2013.

[26] Z. Shelby, "Embedded web services," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 52–57.

[27] P. Rajasekaran, R. Janardhan, and R. Chander, "A smarter toll gate based on web of things," in *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT '13)*, Bangalore, India, 2013.

[28] C. Sammarco and A. Iera, "Improving service management in the internet of things," *Sensors*, vol. 12, no. 9, pp. 11888–11909, 2012.

[29] M. A. Rendón, A. J. Green, E. Aguilera, and P. Almaraz, "Status, distribution and long-term changes in the waterbird community wintering in Doñana, south-west Spain," *Biological Conservation*, vol. 141, no. 5, pp. 1371–1388, 2008.

[30] M. Guillemain, N. Sadoul, and G. Simon, "European flyway permeability and abmigration in Teal Anas crecca, an analysis based on ringing recoveries," *Ibis*, vol. 147, no. 4, pp. 688–696, 2005.

[31] "Dõnana Biological Station," http://www.ebd.csic.es/website1/Principal.aspx.

[32] J. Mora-Merchan, F. Molina, D. Larios, G. Rodriguez, J. Barbancho, and C. León, "Architecture for environmental data access in WSN," in *Proceedings of the International Conference on Data Communication Networking and the International Conference on Optical Communication Systems (DCNET/OPTICS '11)*, pp. 102–106, Seville, Spain, 2011.

[33] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, vol. 48, pp. 364–369, April 2005.

[34] P. Levis, S. Madden, J. Polastre et al., *TinyOS: An Operating System for Sensor Networks*, Springer, 2005.

[35] D. Larios, J. Barbancho, G. Rodríguez, J. Sevillano, F. Molina, and C. León, "Energy efficient wireless sensor network communications based on computational intelligent data fusion for environmental monitoring," *IET Communications*, vol. 6, no. 14, pp. 2189–2197, 2012.

[36] Z. Alliance, *Zigbee Specification*, 2007.

[37] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, ACM, New York, NY, USA, November 2009.

[38] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo, "The collection tree protocol," TEP 123, 2006.

[39] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, pp. 37–45, 2008.

[40] J. Hui and D. Culler, "IPv6 in low-power wireless networks," *Proceedings of the IEEE*, vol. 98, pp. 1865–11878, 2010.

[41] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, 2007.

[42] J. Hui and P. Thubert, *RFC 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, 2011.

[43] "Berkeley Low-power IP stack project," http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip.

[44] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *Draft-Ietf-Core-Coap-13: Constrained Application Protocol (CoAP)*, 2012.

[45] K. Kuladinithi, O. Bergmann, T. Ptsch, M. Becker, and C. Grg, *Implementation of CoAP and Its Application in Transport Logistics*, 2011.

[46] "USB Dongle—IA OEM-DAUB1 2400 by Adaptive Modules Ltd.," http://www.adaptivem2m.com/zigbee-technology/zigbee-usb-dongle.htm.

[47] "Perytons Protocol Analyzers," http://www.perytons.com.

[48] J. E. Higuera and J. Polo, "IEEE 1451 standard in 6LoWPAN sensor networks using a compact physical-layer transducer electronic datasheet," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 8, pp. 2751–2758, 2011.

[49] *IEEE Std 1451.5-2007: IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, 2007.

[50] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained application protocol (CoAP)," Tech. Rep., IETF Secretariat, Fremont, Calif, USA, 2013.