

Data Dissemination in a Wireless Video Surveillance Platform for Elderly Monitoring: Implementation and Experiments

Anis Harfouche¹, Saadi Boudjit²(✉), and Azeddine Beghdadi²

¹ ESI – Ecole nationale Supérieure d’Informatique, 16309 El Harrach, Algiers, Algeria
ba_harfouche@esi.dz

² Université Paris 13, Laboratoire de Traitement et Transport de l’Information L2TI,
99 Av J-Baptiste Clément, 93430 Villetaneuse, France
{boudjit,beghdadi}@univ-paris13.fr

Abstract. Nowadays, the number of elderly people keeps growing and represents a non negligible part of the global population in the world. Consequently, healthcare and monitoring expenses destined to them become more and more important. Indeed, receiving ageing people in dedicated infrastructures with qualified staff costs a lot of money either for them and for governments. Also, a large number of elderly prefer continue to live in their own houses rather than joining those healthcare centers. However, they could be subject to domestic accidents and the latters are often detected after a while. This work aims to setup a efficient wireless video surveillance system to help elderly people who need a permanent assistance while they prefer still living in their houses. Our main objective is to early detect and transmit via Internet any abnormal behavior or domestic accident to assistance services. For this, small cameras embedded on wireless home deployed sensors have been considered. Moreover, a simple and lightweight routing protocol for an optimized data transmission have been proposed. The whole system was implemented on an Arduino based platform on which a set of experiments were conducted.

Keywords: Wireless Sensor Networks (WSNs) · Wireless Multimedia Sensors (WMSs) · Network routing protocols · Platforms · Health databases

1 Introduction

The average age of the population tends to increase and the number of people requiring more or less home intensive monitoring is not small. Indeed, 8.2% of global population in the world are beyond 65 years old [1]. Hence, platforms for remote monitoring of elders may be of interest at different levels. First, it would allow ageing people to stay home which may have an important impact on

their state of mind, a very important element in case of recovery or chronic diseases. Secondly, those platforms are cost effective and use either Bluetooth [2] or ZigBee [3] based radio transmissions to exchange data. In most cases, however, ZigBee is preferred because of its low energy consumption due to the possibility to put the radio in the sleep mode. As depicted in Fig. 1, such platforms are usually composed of sensors with embedded cameras and coordinators called sinks which collect pictures from the network and transmit them to remote databases servers. Sinks are generally linked to a computer or a smartphone/PDA that offer them different transmission technologies for Internet access (WiFi/4G). Pictures on these databases servers can be accessed through simple browsers to allow monitoring staff to have a continuous view of elders environments.

This paper presents our platform details that can be deployed in a common house, permitting the monitoring of several areas (rooms, kitchen, garden, ...) at the same time, by using small non-intrusive devices which route collected data towards sinks. The paper is organized as follows. In Sect. 2, we will present and discuss some previous works related to remote monitoring through WSNs in general and WMSNs in particular. In Sect. 3, we will give more details about our own contribution and some of its key parts. The implementation details of the proposed architecture are presented in Sect. 4. Section 5 concludes the paper and highlights some improvements that can be brought to our contribution as long as future planned works and perspectives.

2 Related Work

Several works on remote monitoring have been proposed in the literature either for health purposes or for safety checking especially for a specific category of persons as elders. The majority of them use scalar sensors embedded on those persons or setted in their environment to collect simple measures as temperature, heartbeat, blood sugar or brain activity. In Oslo University Hospital, researchers implemented and tested a Biomedical Wireless Sensor Network (BWSN) [4] where they integrated six different types of sensors; Wireless Pressure Transducer, DigiVent Pulmonary Air Leakage, CardioPatch ECG sensor, Medical UWB-IR radar, Heart Monitoring Accelerometer, SpO2 & Temperature sensors. Patients which got these sensors could keep a good mobility and met a shorter hospital stay period than usually with this procedure.

In [5,6], a system prototypes that provide distance healthcare at home have been proposed. These systems automatically measure and collect home and body parameters and transmit them to a central server through a public network. The central server analyses retrieved data to form reports about elder's activity and health state, and in case of emergency, automatically sends an alarm to assistance services through voice phone calls, SMS or E-mails.

However, works where monitoring platforms use multimedia aspect are fewer. In [7], the architecture of a monitoring platform based on WMSN is proposed. The monitored persons are equipped with smartphones to be localized by a program agent and the latter sends the localisation to image, sound and depth

based sensors deployed in the environment. The sensors start to track the persons and send data back to the agent. The agent uses machine learning techniques based on usual behavior of those persons to detect any abnormal behavior and launches alerts destined to monitoring staff. For the best of our knowledge no complete implementation of this platform has been made. Moreover, the use and processing of machine learning techniques need a non negligible period of time which will impact the real time aspect of the platform.

Another platform proposed in [8] allows to detect some basic events by making images processing at sensor level. Two consecutive images are compared in order to identify a simple behavior. For example, the sensor is able to detect the event of leaving a room by first noticing nobody in front of the door in a picture followed by a picture with a person in door step and finally a last picture with nobody in front of the door. Obviously, these kind of platforms may be useful only for detecting simple events and would meet more difficulties with more complex events as human falls.

3 Our Contribution

For our solution, we considered the scenario of ageing people living in a house with several areas. For this scenario, several sensors are deployed to maintain a good coverage of all the areas where these people can be and the deployed sensors meet low mobility during network's lifetime. Moreover, we considered a multi-hop sink rooted tree topology for data transmission. Indeed, due to radio range restrictions imposed by ZigBee technology, some sensors may be too far from the sink for a direct data transmission. Therefore, we have proposed a multi-hop routing algorithm which gives the possibility to remote nodes without direct links with the sink to transmit their data to the latter through multi-hop. The general architecture of the proposed system is presented in Fig. 1. Sensor nodes exchange data with the sink which has in addition to its ZigBee interface, a WiFi interface for Internet access without the need for a PC or a PDA. Information about taken pictures are stored in a remote database server and the pictures themselves are stored in a remote file server. This allows the web application we developed, to answer requests coming from end users asking for last pictures related to a specific monitored person. The web application gets information about stored pictures from the database server and then, requests them from the file server. Due to sensor's low calculating capabilities, we are not proposing any image processing made on sensors level. Any processing on pictures can be done at servers level which offer greater processing abilities.

3.1 Data Exchange Within the Network

To take advantage of the static aspect of the nodes within the network, our routing algorithm doesn't use generic 'Hello' and 'Topology control' packets to maintain the topology as it is the case in common routing algorithms for ad hoc networks: CTP [9], MintRoute [10] or MultiHopLQI [11]. The discovery of the

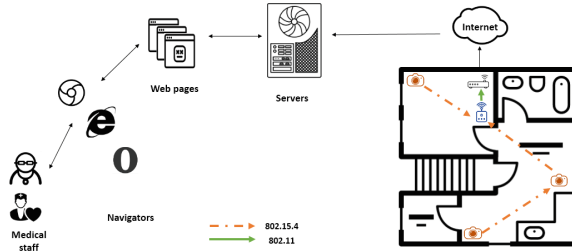


Fig. 1. General platform architecture

whole topology is realized at the initialization phase made at the beginning of the platform deployment. This may help to save network bandwidth and reduce its energy consumption. We propose a sink driven routing protocol to build a sink-rooted tree topology of the network. The sink is the only node that identifies all camera nodes present in the network and it is responsible of sending requests periodically to each camera node in order to ask for a picture. Each camera node has only a local view of the topology which helps it to route data coming from neighborhood towards the sink and requests coming from the sink towards other camera nodes (Fig. 2).

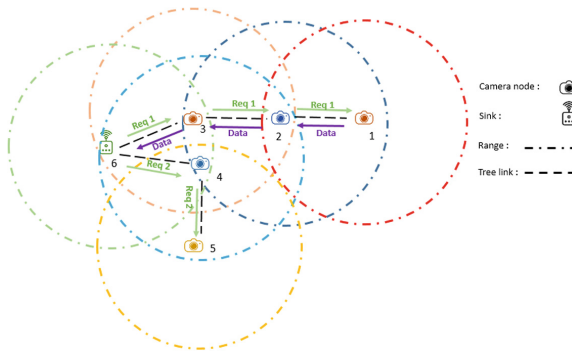


Fig. 2. Example of a tree topology

When camera nodes get launched for the first time, they start to broadcast signed signalisation packets to join the network. This first phase is called Signalisation Phase and it's pseudo-code is given in Algorithm 1. When a camera node receives a signalisation packet from an unknown source, it records the source address of the packet and the one of the node the packet have been relayed from (next hop to the source), then it broadcasts the packet itself. If it receives, however, a signalisation packet generated by an already known source but relayed by a different next hop node, it stores the source address and the new next hop address in an alternate table that would be useful in case of old next hop node

failure. Two modes may be used, the simplest one is to put the first received couple (camera node address, next hop towards it) in the routing table and to store other later discovered routes towards this camera node in the alternate table. A smarter mode, even if it requires more processing, would be to keep in the routing table the link which has the best RSSI (Radio Signal Strength Indicator) and classify other links according to this metric in the alternate table.

Algorithm 1. Camera node Signalisation Phase

```

1: While (Signalisation Timer)
2:   Broadcast signalisation
3:   if (Received signalisation packet)
4:     # source node & relay node refer to the packet
5:     if (Unknown source node)
6:       Add_route(source node, relay node)
7:     elseif (Unknown (source node, new relay node))
8:       Add_alternate(source node, new relay node)
9:     relay node = me
10:    Broadcast the packet

```

After a period of time, all camera nodes which are not isolated from the network (out of range of all other nodes) are listed at the sink level. The sink keeps the destination address of each known camera node in the network and the next hop to reach it. The next hop is the node address from which the signalisation packet of the destination address have been received. The sink signalisation phase is exactly the same as for camera nodes except that the sink doesn't broadcast the signalisation packets it receives.

We use at each camera node a routing table to route requests and data packets. An entry of the table contains the address of a camera node, next hop to reach it and next hop to route its data towards the sink. For example, the tables of nodes 2 and 4 in Fig. 2 are described in Fig. 3.

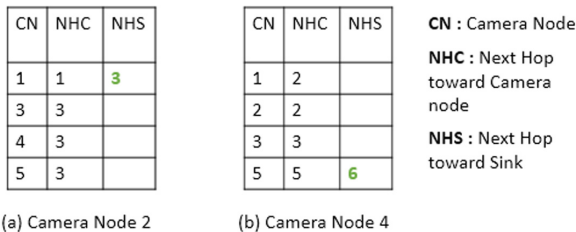


Fig. 3. Routing tables

First and second fields of each entry of the table are filled during Signalisation Phase, third field is filled when a sink request is received. A similar table is used

at the sink with only the first and second fields, the third being useless. At the end of the Signalisation Phase, a second phase called Collection Phase begins. The sink starts to periodically send, in a Round Robin order, unicast requests to camera nodes it has recorded in the first phase to ask for pictures. The pseudo-code of this phase is given in Algorithm 2. This technique is a very simple way to avoid additional delays generated by packets losses due to channel multiple access. Indeed, the sink is responsible of organizing data production in the network, which avert getting several data sources transmitting at the same time.

Algorithm 2. Sink Collection Phase

```

1: if (Cycle Timer Expires)
2:   foreach: CN in RoutingTable
3:     Send request to CN
4:     Wait for data from CN
5:     if (receiving data)
6:       while (receiving data)
7:         Concatenate in SD file
8:         Upload file's data to database
9:         Upload file to FTP server
10:    elseif (timer expires)
11:      if (NbRetransmissions reached)
12:        if (alternate route exists)
13:          Swap it to routingTable
14:        else
15:          Retransmit request packet
16:          Go to 5
  
```

At the reception of a request packet, a node verifies if it is addressed to it. If it isn't, it records the address of the node from which it got the request, this one will be the next hop towards the sink for data packets coming from the camera node specified as destination address in the request packet. Then, this intermediate node forwards the request packet to the next hop it has recorded during Signalisation Phase towards the destination camera node. When receiving a request packet addressed to it, a camera node takes a picture and starts to send data packets to sink via the node it got the request packet from. This way, request packet and data packets follow the same route (in opposite directions) for each camera node. The pseudo-code of this process is given in Algorithm 3.

If the sink (respec. a camera node) generates (respec. forwards) a request without getting back data after a period of time, it retransmits the request packet. If after several attempts no data packet is received, it suspects a failure of the next hop. If a route towards the same destination camera node is found in the alternate table, it replaces the one in the routing table and get deleted from the alternate table. If no alternate route is found, the request packet is locally dropped. In Fig. 4, we give an example of the backup scenario at sink level in case of failure of the node 3 in Fig. 2.

Algorithm 3. Packet Processing at Camera Node

```

1: if (Received packet via a relay node)
2:   switch (packet type)
3:     Request:
4:     # Source & relay & destination refer to the packet
5:     if(destination == me)
6:       Take snapshot
7:       Send fragments to the source via relay node
8:     else
9:       NHS = relay in routingTable where CN == destination
10:      Forward request to NHC in routingTable where CN == destination
11:     if (timer expires)
12:       if (NbRetransmissions reached)
13:         if (alternate route exists)
14:           Swap it to routingTable
15:         else
16:           Retransmit request packet
17:         Go to 11
18:     Data:
19:     Forward to NHS in routingTable where CN == source

```

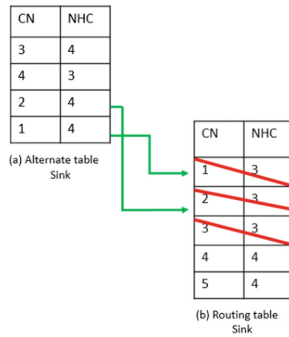


Fig. 4. Sink’s routing and alternate tables

3.2 Picture’s Remote Display

When the sink starts receiving data packets, it concatenates them in a picture file located in the MicroSD memory because of the lack of storage space in the memory of the node. The MicroSD acts like a buffer for the received pictures before being uploaded to remote servers. At the reception of all packets composing a picture, the sink which established a connexion to database and FTP servers at initialization phase, inserts into the database images table a novel entry where it specifies the identification of the camera node which captured the picture (ZigBee address in the network), the identification of the monitored person and the date/time of picture’s capture. This 3-tuple identifies in a unique manner a picture as shown in Fig. 5. The second operation for the sink is to upload the picture to the FTP server.

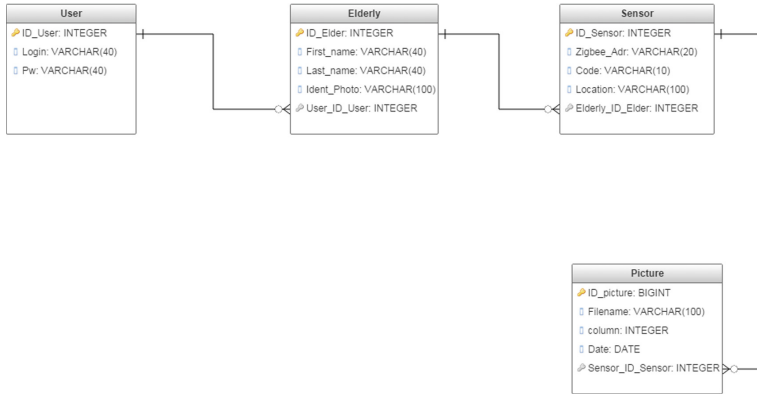


Fig. 5. Database scheme

When a user asks through a web application for pictures related to a specific person, a request is addressed to the database to get all file names of pictures related to this person. For each file name, the picture is retrieved from the FTP server and then displayed. The user can address more detailed requests by specifying an interval between two dates or by asking only for pictures of a specific area of the monitored structure.

4 Implementation

The sensors used in our platform are Arduinos Mega 2560 equipped with cameras. We used TTL Serial cameras and Wireless proto Shields with XBee S1 senders to handle ZigBee transmissions between camera nodes. It was possible for us to use a 32 bits long addresses for the nodes, however, due to the nature of the use case of our platform we opted for 16 bits addresses. The sink doesn't integrate a camera but has a Wireless WiFi Shield to connect to an access point and upload collected data into servers. This shield contains a MicroSD slot which is used by the sink to store a picture while the fragments composing it are being received. When a picture is fully received, the sink transmits its information to remote database and then uploads it into remote FTP server. We have used three servers, a MySQL database server V5.7 to store information about end-users accounts, monitored persons as long as received pictures and sensors details. A FileZilla server V0.9.49 used to store photos of monitored persons and pictures sent by sensors. Finally, a JEE based web application running on an Apache Tomcat V8.0 server. The web application communicates with FTP and database servers to retrieve pictures and information asked for by the end-user. We tried several scenarios for our network topology to retrieve pictures from nodes up to three hops far from the sink as in Fig. 6. We used a WiFi LAN to access to remote servers which were running on a laptop. The web browser to access to the web application was used on an end-user laptop. Finally, a smartphone was used as an access point.

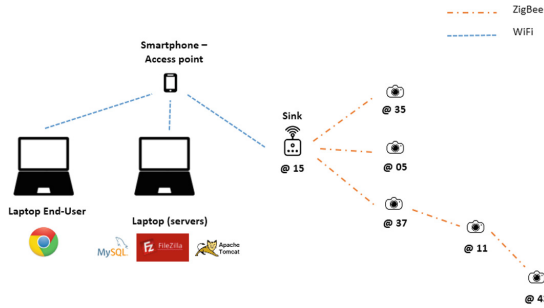


Fig. 6. Scenario scheme

At the initialization phase, the sink connects to the LAN through the access point and creates a connexion with the remote database and FTP servers. After the signalisation phase which runs for 2 min, it starts to receive data from camera nodes and to upload it to the servers. We didn't implement yet node's failure detection scheme based on alternate tables. The sink detects a non responding node if it doesn't receive data from it after a period of time following a request transmission, after what, a new request is then retransmitted.

Using NTP to keep a continuous correct time at Arduino level needs too frequent refresh and, hence, consumes more energy. In order to still be able to have the time and the date where pictures have been taken, we use a Python script at servers level to change the file name of a picture to insert time and date of its reception at the server. It also exchanges the ID of the sensor in the file name with an abbreviation of its location, which is more meaningful for the end-user.

5 Conclusion and Future Works

In this paper, we first proposed a routing algorithm for images collection in a WMSN adapted to a specific case of use which is elders monitoring in domestic areas. Then we implemented our algorithm in a generic platform based on Arduino Mega 2560 multimedia sensors. The presented solution may be seen as an alternative for regular monitoring platforms based on fixed cameras. It may offer some advantages compared to the latter in terms of mobility, simplicity of network expansion and thus covered area, and continuity of service compared to fixed cameras in case of energy breakdown (power cut off).

Finally, we have identified several ideas which could improve our platform in our future works. First, the implementation at server level of an automatic scheme for abnormal behavior detection and alert generation such as elderly falls. We will also be working on empirical studies to determine the best size of picture frames transmitted within the network to achieve the best delivery time. It would also be interesting to consider the use of scalar sensors as long as multimedia sensors; the two types being complementary for a global health and safety monitoring.

References

1. World population in 2013. <http://www.worldometers.info/fr/population-mondiale/#age>. Accessed June 2016
2. Bluetooth Technology - Bluetooth Special Interest Group (SIG). www.bluetooth.com
3. Zigbee Alliance - 802.15.4 Standard. www.zigbee.org
4. Eirik, N., et al.: Biomedical Wireless Sensor Network, Phase II. http://www.imatis.com/imatis/wireless-sensor-healthcarehomecare_files/BWSN%20II%20Technical%20Project%20Report_March2010.pdf. Accessed June 2016
5. Townsend, K.: Recent advances and future trends on low power wireless systems for medical applications. In: 2005 Proceedings of Fifth International Workshop on System-on-Chip for Real-Time Applications, pp. 476–478 (2005)
6. Boudra, H.: Un prototype de systeme de telesurveillance medicale base sur les capteurs et les reseaux de capteurs sans fil (2014)
7. Jin, S.-Y., Jeong, Y.-S., et al.: An intelligent multi-sensor surveillance system for elderly care. *Smart CR* **2**(4), 296–307 (2012)
8. Al-Marakeby, A.: Camera-based wireless sensor networks for E-Health. *Adv. Res. Comput. Commun. Eng.* **2**, 4757–4761 (2013)
9. Fonseca, R., Gnawali, O., et al.: The collection tree protocol (CTP). *TinyOS TEP* **123**, 2 (2006)
10. Woo, A., Tong, T., et al.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems. ACM (2003)
11. Tolle, G., Levis, P., Buoadonna, P., Woo, A., Polastre, J.: MultiHopLQI protocol (2007)