# Deploying a Pool of Long-Range Wireless Image Sensor with Shared Activity Time

Congduc Pham
University of Pau, LIUPPA Laboratory
congduc.pham@univ-pau.fr

***Abstract*—Long-range radio technologies can connect remote sensors/IoT devices without the complex and costly deployment of relay nodes. In case of image sensors, the larger amount of data to transmit can rapidly results in a radio activity time greater than the duty-cycle limit allowed per hour. We propose an activity time sharing mechanism for a pool of image sensors deployed by a single organization. We propose to overcome the tight 36s/hour radio activity of a device by considering all the individual activity time in a shared/global manner. Devices that need to go beyond the activity time limitation can borrow activity time from other devices to provide better surveillance service guarantee. Providing simple, low-cost, long-range connectivity with the possibility of sharing activity time can make large-scale data-intensive applications such as visual surveillance a reality. The proposition is implemented on our low-cost image sensor platform and preliminary tests show that the proposed mechanism is fully functional.**

***Index Terms*—Wireless image sensor networks, long-range transmission, long-range surveillance.**

## I. Introduction

The monitoring capability of Wireless Sensor Networks (WSN) make them very suitable for large scale surveillance systems. Going beyond simple physical measures such as temperature or luminosity the possibility to provide visual information can greatly enhance a number of surveillance applications. In this paper, we address wireless image sensor network for fully autonomous visual surveillance. However, the deployment of such sensors in a large scale is still held back by technical challenges such as short communication distances. Using the telco mobile communication infrastructure is still very expensive (e.g. GSM/GPRS, 3G/4G/LTE) and not energy efficient for autonomous devices that must run on battery for months. During the last decade, low-power but short-range radio such as IEEE 802.15.4 radio have been considered by the WSN community with multi-hop routing to overcome the limited transmission range. While such short-range communications can be realized on smart cities infrastructures, it can hardly be generalized for the large majority of surveillance applications that need to be deployed in isolated or rural environments. We propose an extreme long-range, low-power, low-bandwidth radio version of our image sensor. Using recent modulation techniques where the long transmission distance (several kilometers in NLOS conditions) can be achieved without relay nodes greatly reduces the complexity of deployment and data collection. This statement is shared by many Machine-to-Machine (M2M) and Internet of Thing (IoT) actors and the concept of Low Power Wide Area Networks (LPWAN) is gaining incredible interest. Some low-power long-range technologies such as Sigfox are still operator-based. However, other technologies such as LoRa™ proposed by Semtech radio manufacturer can be privately used and deployed following the recent LoRaWAN™ specifications [1]. This last solution has the following advantages over traditional short-range technologies:

1) avoid relying on operator-based communications; no subscription fees;
2) remove the complexity and cost of deploying a multi-hop infrastructure;
3) can offer out-of-the-box connectivity facilities.

Fig. 1 shows a typical extreme long-range 1-hop connectivity scenario to a long-range base station (LR-BS) which is the single interface to Internet servers. According to tests performed both by Libelium and Semtech, one can at least expect 20km range in LOS condition and 2km in NLOS, urban area where the RF signal has to travel through several buildings [2], [3].

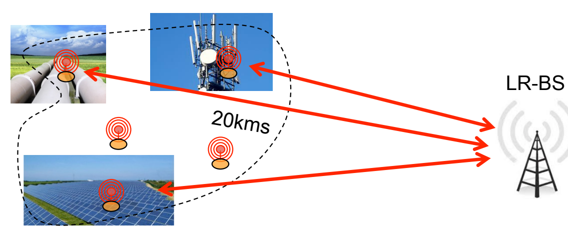


Fig. 1. Extreme long-range application

The flexibility of long-range transmission comes at the cost of stricter legal regulations. For instance, electromagnetic transmissions in the sub-GHz band of Semtech's LoRa technology falls into the Short Range Devices (SRD) category. In Europe, the ETSI EN300-220-1 document [4] specifies various requirements for SRD devices, especially those on radio activity. Basically, a transmitter is constrained to 1% duty-cycle (i.e. 36s/hour) in the general case. This duty cycle limit applies to the total transmission time, even if the transmitter can change to another channel.

Our image sensor works with raw 128x128 8-bbp gray scale image and integrates a simple-differencing technique for image change detection. The image can be compressed with various quality factors for reducing the bandwidth usage. This is especially important with long-range radio technologies where the longer distance is made possible by using advanced spread spectrum modulation techniques at the cost of operating at very low data rates, i.e. less than 10kbps in most cases. Still, the transmission of an image at medium quality can easily be done in less than 36s. However, in case multiple images need to be sent in a short time interval, the activity regulation of 36s/hour can rapidly be very limiting.

In this paper we address the case of deploying a pool of long-range image sensors, managed by a single organization. We propose to overcome the tight 36s/hour radio activity of a device by considering all the sensor's individual activity time in a shared/global manner. Devices that need to go beyond the activity time limitation can borrow some from other neighbors.

The rest of the article is organized as follows. Section II describes the long-range image sensor node and image processing performance measures. Section III details the long-range technology and shows how the developped long-range image sensor can be used to build an out-of-the-box surveillance system. In Section IV we will present the proposed mechanism that considers all the sensor node's individual activity time in a shared/global manner. We will also discuss on the cost of adding the activity time sharing mechanism. We conclude in Section V.

## II. Review of the long-range image sensor

There are a number of image sensor boards available or proposed by the very active research community on image and visual sensors: Citric [5], WiCa [6], Eye-RIS [7], Panoptes [8], CMUcam4 and CMUcam5/PI-XY [9] to name a few. All these platforms and/or products are very good but are either based on ad-hoc development of the visual part (i.e. development of a camera board with dedicated micro-controller to perform a number of processing tasks) or designed with powerful micro-controller/Linux-based platforms or do not propose efficient image encoding and compression scheme adapted to low-resource devices. Our motivations in building our own image sensor platform for research on image sensor surveillance applications are: ($i$) to propose an off-the-shelf solution for maximum reproducibility, flexibility in programming and design; ($ii$) to develop and experiment an efficient image compression scheme running on host micro-controller (no additional nor dedicated micro-controller) which addresses the problem of resource limitations of sensor nodes while producing a highly-tolerant to packet losses bit stream.

### A. Hardware components

We use Arduino boards with the CMOS uCamII camera from 4D systems. The uCamII is shipped with a $56^o$ angle of view lens but we also use $76^o$ and $116^o$ lenses. The uCam is connected to the Arduino board through an UART interface at 115200 bauds. The uCamII is capable of providing both raw and JPEG bit streams but we are not using this last feature as it is impossible from the delivered JPEG bit stream to build a packet stream tolerant to packet losses. As a result, we retrieve raw 128x128 8-bpp grey scale images from the uCamII then we operate image compression on the Arduino board. We proposed 2 versions of the image sensor. One is built on an Arduino Due board and the other on the Arduino MEGA2560. The Arduino Due is a micro-controller board based on the Atmel SAM3X8E ARM Cortex-M3 running at 84MHz with 96KB of SRAM memory. The MEGA2560 features an ATmega2560 at 16Mhz and has only 8KB of SRAM memory. The Arduino Due would represent a medium-end platform while the MEGA2560 is a low-end platform. Fig. 2 shows the developed image sensor built from off-the-sheves components. Several cameras could also be attached.

Initially, an XBee S1 module from Digi provided an IEEE 802.15.4 short-range connectivity. In the long-range version, a multi-protocol radio shield from Libelium allows for connecting both the XBee S1 module and a long-range radio module consisting in a Libelium SX1272 LoRa module [3] built upon Semtech's LoRa SX1272 chip [10] that implements a proprietary spread spectrum technology in the licensed-free sub-GHz 860MHz-1020MHz frequency band. Although the presence of both radio technologies can lead to some very interesting networking perspectives, we will focus in this paper on the LoRa new features and performances. Compared to legacy modulation techniques, Semtech's LoRa permits an increase in link budget and increased immunity to in-band interference. In addition, we only use the Due platform for the new long-range development because although the very limited amount of memory on the MEGA was a challenge to solve for the image encoding process, the Due is much faster while also being more energy efficient (see [12]).

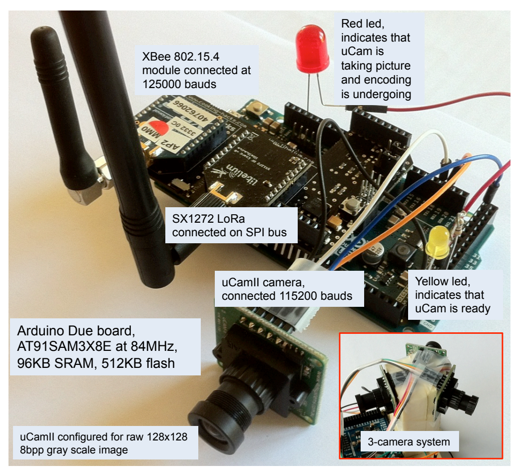


Fig. 2. Image sensor built with Arduino (Due or MEGA) and uCam camera

### B. Image processing tasks

*1) Image change detection:* The sensor implements a "simple-differencing" method that also takes into account slight modifications in luminosity. We tested the image change detection in intrusion detection tests and were able to systematically detect a single person intrusion at more that 25m without any false alert. Note that traditional infrared presence sensors (PIR) can not provide detection at that distance. In addition, the image change detection mechanism can be used and tuned to detect changes in close-up views for various phenomenon detection: cracks, leakages,...

*2) Image compression method:* On image change, the sensor can transmit the image to a base station. We use an optimized encoding scheme proposed in [13] which features the 2 following key points: $(i)$ image compression must be carried out by independent block coding in order to ensure that data packets correctly received at the sink are always decodable and, $(ii)$ de-correlation of neighboring blocks must be performed prior to packet transmission by appropriate interleaving methods in order to ensure that error concealment algorithms can be efficiently processed on the received data. For these reasons the encoded bit stream is particularly tolerant to packet losses which is highly important for very low-bandwidth and high-latency technologies such as long-range radios. Additionally, a tuning parameter, called Quality Factor (Q), provides a compression ratio/energy consumption trade-off that can further be used to optimize transmission time.

Fig. 3(left) shows the original raw 128x128 image taken with the image sensor and encoded with various quality factors: Q=90 (high quality), Q=50 (medium quality) and Q=10 (low quality). The compressed image size, the compression ratio (in bracket) the number of generated packets and the PSNR compared to the original image are shown.

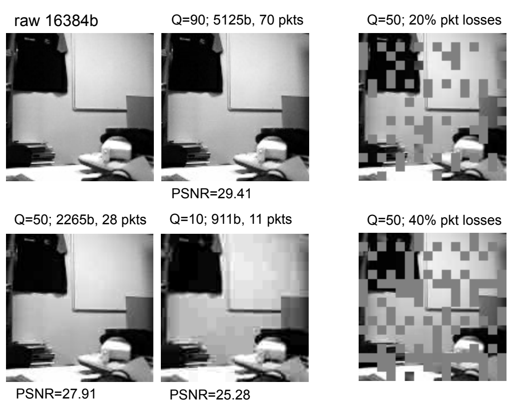


Fig. 3. 128x128 image taken by the image sensor, various quality factor.

The number of packets depends on the image maximum segment size (IMSS) allowed per packet. For Fig. 3(left) we set it to 90B. The produced packet size will slightly vary according to the packetization process. 7 bytes need to be added to the image payload: source image sensor address (2B), packet sequence number (1B), quality factor (1B), image payload size (1B) and offset of the first block of image data in the packet (2B). Therefore, with an IMSS of 90B, each packet is close to 100B. Fig. 3(right) shows the impact of packet losses (IMSS was set to 90B) on the image quality.

The 90-byte limit was initially defined for 802.15.4 XBee radio where the maximum MAC payload size is 100B. Semtech's LoRa radio can handle up to 255B payload in variable packet size mode. As the cost of packet sending is usually reduced using larger packets it is more beneficial to increase the IMSS close to the radio limit. When removing Libelium API header bytes (5B) and the 7-byte image header, 243B remain available. This will be rounded down to 240B as 3 bytes will be used by our proposed protocol.

Obviously a tradeoff should be found for the IMSS between the gain in sending time and the impact of a packet loss on the image quality. For instance, with IMSS=90, the image encoded at Q=50 produces 28 packets while using IMSS=240 only produces 6 packets, thus giving a packet loss percentage of nearly 17% per packet. This issue will however not be addressed in this paper and link quality may be monitored to adapt IMSS accordingly.

## C. Image processing performance measures

Fig. 4 summarizes for various quality factors the image encode time (column E). Then the number of produced packets (column N), the encoded image size with the compression ratio (column 1) and the packetization time (column P) are indicated for IMSS=90 and IMSS=240. All these measures are taken without transmission of packets. The time to read the raw image data from the uCam is also shown in column R (1512ms) and it actually does not depend much on the uCam-Arduino connection baud rate (here 115200 bauds) because the limitation is mainly due to memory read operations from the Arduino UART ring buffer. R+E+P represents the elapsed time between the snapshot taken by the camera and the time all the packets of the encoded image are produced (once again without transmission).

| | | | MSS=90 | | | MSS=240 | | |
|---|---|---|---|---|---|---|---|---|
| R | | E | N1 | S1 | P1 | N2 | S1 | P2 |
| reading time from ucam | Quality Factor Q | encode time | number of packets | size in bytes (compression ratio) | packetization time | number of packets | size in bytes (compression ratio) | packetization time |
| 1512 | 100 | 387 | 159 | 10549 (1.55) | 629 | 52 | 10617 (1.54) | 465 |
| 1512 | 90 | 515 | 75 | 5476 (2.99) | 204 | 25 | 5552 (2.95) | 174 |
| 1512 | 80 | 513 | 54 | 4031 (4.06) | 135 | 19 | 4091 (4) | 121 |
| 1512 | 70 | 520 | 42 | 3272 (5) | 106 | 15 | 3289 (4.98) | 101 |
| 1512 | 60 | 512 | 35 | 2800 (5.85) | 94 | 13 | 2832 (5.78) | 89 |
| 1512 | 50 | 504 | 30 | 2447 (6.69) | 83 | 11 | 2458 (6.66) | 82 |
| 1512 | 40 | 519 | 28 | 2196 (7.46) | 78 | 10 | 2196 (7.46) | 76 |
| 1512 | 30 | 519 | 23 | 1817 (9.01) | 68 | 8 | 1824 (8.98) | 69 |
| 1512 | 20 | 518 | 17 | 1428 (11.47) | 60 | 7 | 1425 (11.49) | 62 |
| 1512 | 10 | 519 | 11 | 920 (17.8) | 50 | 4 | 922 (17.77) | 55 |
| 1512 | 5 | 519 | 7 | 555 (29.52) | 44 | 3 | 553 (29.62) | 48 |

Fig. 4. Image processing performance. All times are in ms.

Increasing the IMSS to 240B generally decreases the packetization time (column P2 vs P1) but the total encoded file size is a bit larger. However for Q=5 to Q=30, the packetization time using IMSS=240 is larger than for IMSS=90. This can be explained by the reduction in the produced packet number difference between both case of IMSS. As the packetization process is an incremental process taking IMSS as an upper limit, the higher the IMSS, the higher the number of cycles needed to fill the packet. Then we can notice that for values of Q up to 70, the difference is below 5ms. Since values for Q on low data rate radio should be small (e.g. 20 or lower), we can therefore consider that the IMSS size has no significant impact on the encoding and packetization time.

## III. Long-range transmission

### A. Long-range Semtech LoRa technology

Electromagnetic transmissions in the sub-GHz band of Semtech's LoRa technology falls into the Short Range Devices (SRD) category. In Europe, the ETSI EN300-220-1 document [4] specifies various requirements for SRD devices, especially those on radio activity. Basically, transmitters are constrained to 1% duty-cycle (i.e. 36s/hour) in the general case. This duty cycle limit applies to the total transmission time, even if the transmitter can change to another channel. Actually, the relevant measure is the time-on-air (ToA) which depends on the 3 main LoRa parameters: BW, CR and SF. BW is the physical bandwidth for RF modulation (e.g. 125kHz). Larger signal bandwidth allows for higher effective data rate, thus reducing transmission time at the expense of reduced sensitivity improvement. CR, the coding rate for cyclic error coding to perform forward error detection and correction. Such error coding incurs a transmission overhead and the lower the coding rate, the higher the coding rate overhead ratio, e.g. with CR=4/5 the overhead ratio is 1.25 which is the minimum value. Finally SF, the spreading factor that can be set from 6 to 12. The lower the SF, the higher the data rate transmission but the lower the immunity to interference thus the smaller is the range.

The Libelium LoRa radio module includes the Semtech's SX1272 radio chip and the Libelium programming library defines 10 so-called LoRa modes that use various combinations of BW, CR and SF settings. We use the formula given by Semtech in [10] to compute the ToA for all LoRa mode defined by Libelium. This is illustrated in Fig. 5. Mode 4 to 6 provide quite interesting tradeoffs for longer range, higher data rate and immunity to interference. Fig. 6 shows graphically the ToA.

| | | | | time on air in second for payload size of | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LoRa mode | BW | CR | SF | 5 bytes | 55 bytes | 105 bytes | 155 Bytes | 205 Bytes | 255 Bytes |
| 1 | 125 | 4/5 | 12 | 0.95846 | 2.59686 | 4.23526 | 5.87366 | 7.51206 | 9.15046 |
| 2 | 250 | 4/5 | 12 | 0.47923 | 1.21651 | 1.87187 | 2.52723 | 3.26451 | 3.91987 |
| 3 | 125 | 4/5 | 10 | 0.28058 | 0.69018 | 1.09978 | 1.50938 | 1.91898 | 2.32858 |
| 4 | 500 | 4/5 | 12 | 0.23962 | 0.60826 | 0.93594 | 1.26362 | 1.63226 | 1.95994 |
| 5 | 250 | 4/5 | 10 | 0.14029 | 0.34509 | 0.54989 | 0.75469 | 0.95949 | 1.16429 |
| 6 | 500 | 4/5 | 11 | 0.11981 | 0.30413 | 0.50893 | 0.69325 | 0.87757 | 1.06189 |
| 7 | 250 | 4/5 | 9 | 0.07014 | 0.18278 | 0.29542 | 0.40806 | 0.5207 | 0.63334 |
| 8 | 500 | 4/5 | 9 | 0.03507 | 0.09139 | 0.14771 | 0.20403 | 0.26035 | 0.31667 |
| 9 | 500 | 4/5 | 8 | 0.01754 | 0.05082 | 0.08154 | 0.11482 | 0.14554 | 0.17882 |
| 10 | 500 | 4/5 | 7 | 0.00877 | 0.02797 | 0.04589 | 0.06381 | 0.08301 | 0.10093 |

Fig. 5. Time on air for various LoRa modes as payload size is varied.

Listen Before Talk (LBT) along with Adaptive Frequency Agility (AFA) can be used to go beyonds the 1% duty-cycle limit but then additional restrictions are introduced: the Tx on-time for a single transmission cannot exceed 1s. If this 1s limit is respected, then the transmitter is allowed to use a given channel for a maximum Tx on-time of 100s over a period of 1 hour for any 200kHz bandwidth. The advantage is that using AFA to change from one channel to another, longer accumulated transmission time is possible. One drawback is that if the image sensor works following the LBT+AFA scheme, then in Fig. 5 all ToA greater than 1s cannot be used. If we look at mode 4, then the maximum payload that can be used is 114B, see Fig. 6, therefore the IMSS can be set to 102B and 100 packets can be sent before changing channel will be required. However, if we want maximum range by using mode 1 for instance, we can see that most payload sizes have ToA greater than 1s!

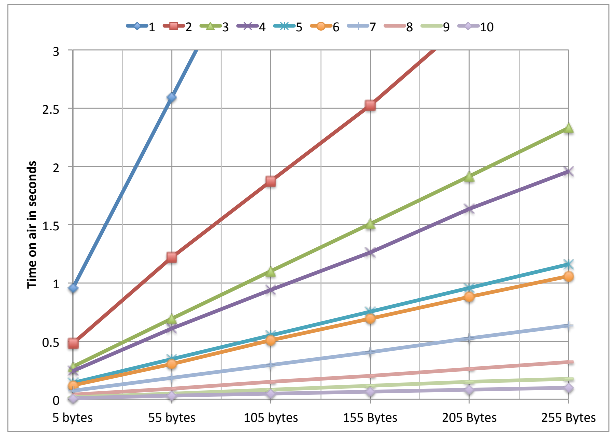


Fig. 6. Time on air for various LoRa mode as payload size is varied.

### B. Out-of-the-box long-range visual surveillance

Each image sensor should be configured to send in a given channel and have an 8-bit address. Then a star-topology as shown in Fig. 1 would allow the LR-BS to collect images from remote nodes. Although a mesh topology can be built with direct communications between image sensors (for alerting purpose for instance) we will not address this issue in the paper to focus on a centralized surveillance system. The LR-BS can be built for outdoor usage with rugged casing to cope with harsh outdoor conditions and simply be an embedded Linux board such as Raspberry PI or Intel Galileo™ with a WiFi or Ethernet connection to transfer the received data. These platforms already support the Libelium multi-protocol radio shield to plug the Libelium SX1272 module. For demonstration purpose, we built the LR-BS with an Arduino MEGA equipped with a Libelium LoRa module acting as a transparent RF-USB-serial bridge: everything that is received on the radio interface will be written to the Arduino serial port which is connected to a Linux machine (laptop). The LR-BS can have an address to realize address filtering when remote nodes send in unicast mode, or can run in promiscuous mode to accept from all nodes.

The image sensor software is designed to work autonomously once powered on: on startup a first image for each camera will be taken to serve as the reference image, then periodic image change detection will send images to the LR-BS in case of image change. At the LR-BS, a display program will continuously waits for valid image packets from the RF-USB-serial interface. Received image packets from various image nodes will be handle accordingly and displayed after a display timer. Each image node has an 8-bit source address set at compilation time, that is included in the image packet header. The camera index is included as well in case of multi-camera nodes (up to 4 cameras with Libelium SX1272 module on SPI bus as all 4 serial ports are available). The screenshot illustrated in Fig. 7 shows a 1-camera and a 3-camera image sensors deployed in our university building.

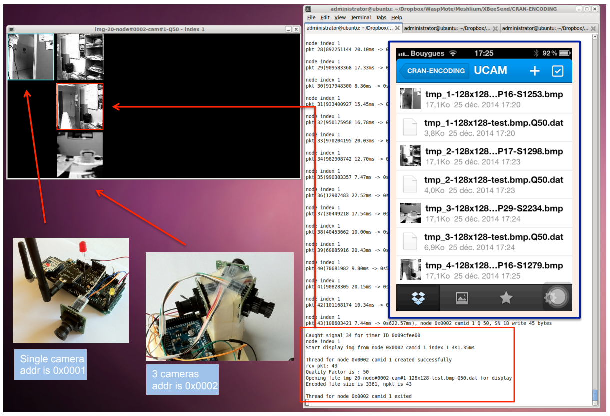


Fig. 7. Transmitted images displayed at a base station

The 1-camera sensor is configured with source address 0x01 while the 3-camera system has source address 0x02. The display tool will dynamically discover new nodes and assign for each node a column index in increasing order. Here column index 0 (left-most) is for node 0x01. Node 0x02 has column index 1. As node 0x02 has 3 cameras, the image taken by each camera appears on a different line. The top line is for camera 0. We can also see how the display window indicates which image is the last one, and which image for a given image node is the last received one: the blue frame indicates for a given image node which image is the last received while the red frame (only one red frame at any time) is the last received image. Additionally, received images stored in a folder could be shared in real-time with a smartphone through a cloud application such as Dropbox™ as shown in figure 7 with the Dropbox application for iPhone™.

### C. Long-range image transmission tests

Extensive LoRa long-range tests have been performed by both Libelium and Semtech and more than 20kms could be achieved in LOS conditions [3]. Semtech's tests also included transmission from pits. We show below some image transmission tests we did in the university area, close to city downtown in a dense urban area with many buildings (NLOS) between the image source and the receiver located in front of the science faculty building. Both receiver and transmitter are at 1.5m height. Libelium LoRa mode 4 is used. We set the IMSS to 240 bytes and the quality factor to 20. Between 8 and 12 packets per image were generated: ToA for a single image is then between 15s and 23s. With mode 4, we could not received at 1010m as indicated in Fig. 8. 1 packet was lost in image 5. By using mode 1 which provides the longest range, we could increase the distance to 1.8km in a very dense urban area.

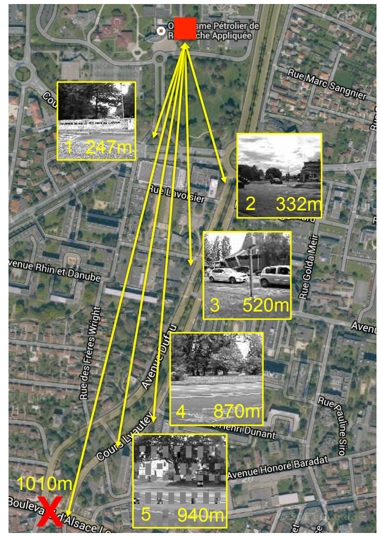


Fig. 8. Long-range tests

## IV. Activity time sharing

As the image data is more voluminous than traditional scalar measures, there are several ways to adapt the image sensor transmission strategy to long-range constraints. For instance, before sending the image, the ToA for all the expected produced packets could be computed (using the current LoRa mode settings) and compared to the remaining activity time in this period. If the computed ToA is greater than the remaining activity time, then the image sensor can use a lower quality factor to reduce the encoded image size, thus reducing the number of packets. Note that even if the increase of ToA is almost linear with respect to the real payload, for a small IMSS there will be more packets generated then more bytes used by various protocol headers. Therefore, increasing the IMSS is also a way to reduce the total ToA at the cost of higher impact of packet loss or packet error.

Preventive measures can also try to reserve some activity time to be able to send an image at the lowest quality at any time in an 1 hour period (e.g. reduce in a preventive way the quality factor). However, in case of major change in the environment, several images may need to be sent in a short time interval, making all previously mentioned measures insufficient. For these scenarios, we propose an additional mechanism which considers the activity time of all deployed long-range devices in a shared manner: an organization deploying a pool of $n$ long-range devices can use up to a Global Activity Time of $G_{AT} = n \times S_{AT}$ per hour, where $S_{AT} = 36000ms$ (time is expressed in ms to avoid complex floating point variable coding). Then, the basic idea is to allow each long-range device to use up to $G_{AT}$ and know its evolution over the 1-hour period.

### A. Proposed activity time sharing mechanism

There could be 2 approaches for devices to update their knowledge of $G_{AT}$: a decentralized or a centralized approach. In a decentralized approach, devices can listen for radio activity and compute the ToA of packets that are sent by other devices to decrease $G_{AT}$ accordingly. However, since devices usually go to sleep mode most of the time to save energy, listening to radio activity can not be done without increasing dramatically the energy consumption. Therefore, we propose the usage of a centralized approach where the LR-BS updates $G_{AT}$ on reception of packets from remote devices and will broadcast new values for $G_{AT}$ at appropriate moment as it will be explained later on. The centralized approach is quite well adapted to the way long-range infrastructures are working, with the LR-BS acting as the single point of interface from remote devices to Internet servers. We propose the following simple centralized radio activity sharing approach:

1) **Initialization**

1a) all deployed long-range devices $D_i$ sharing their activity time initially register (REG packet) with the LR-BS by indicating their local Remaining Activity Time $l^i_{RAT0}$. The LR-BS stores all $l^i_{RAT0}$ in a table (the last $l^i_{RAT0}$ value is also saved), computes $G_{AT}$ and broadcasts (INIT packet) both $n$ (the number of devices) and $G_{AT}$, see Fig. 9(left). For sake of simplicity we assume that all devices start at $t_0$ and that they share 100% of their local activity time. It is possible to handle different startup time and a fraction of local activity time (by indicating a $l^i_{RAT0} < S_{AT}$ in the registration message). Note that this step is performed periodically every hour.

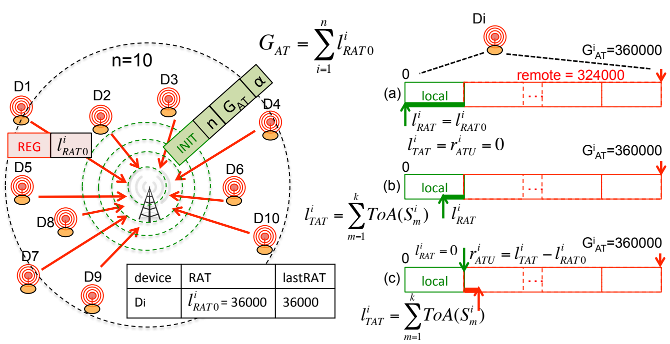


Fig. 9. Left: initialization. Right: device's local and remote activity time

1b) on reception of $n$ and $G_{AT}$ from INIT message each device $D_i$ can consider an initial (and local) $G^i_{AT} = l^i_{RAT0} + \sum_{j=1, j\neq i}^{n} l^j_{RAT0}$, as shown in Fig. 9(right)(a). $D_i$ also sets its local Remaining Activity Time, $l^i_{RAT}$ (the green bar), to $l^i_{RAT0}$ and both its local Total Activity Time, $l^i_{TAT}$, and its remote Activity Time Usage, $r^i_{ATU}$, to 0. It is possible to limit the $G_{AT}$ ratio allowed for usage to $\alpha \times G^i_{AT}$. We assume here that $\alpha = 100\%$.

2) **Device $D_i$ wants to send a DATA packet $k$ of size $S^i_k$**

2a) $D_i$ computes $ToA(S^i_k)$.

2b) if $l^i_{TAT} + ToA(S^i_k) > \alpha \times G_{AT}$ then ABORT.

2c) $D_i$ updates $l^i_{TAT} = l^i_{TAT} + ToA(S^i_k)$ and $l^i_{RAT} = l^i_{RAT} - ToA(S^i_k)$, see Fig. 9(right)(b).

2d) if $l^i_{TAT} > l^i_{RAT0}$ then $D_i$ sets $l^i_{RAT} = 0$ and $r^i_{ATU} = l^i_{TAT} - l^i_{RAT0}$ (the red bar), see Fig. 9(right)(c).

3) **LR-BS receives a DATA packet $k$ from $D_i$ of size $S^i_k$**

3a) LR-BS computes $ToA(S^i_k)$ and updates for device $D_i$ $l^i_{RAT0} = l^i_{RAT0} - ToA(S^i_k)$.

3b) when last packet or timeout from $D_i$ compute $AT^i = l^i_{RAT0} - lastl^i_{RAT0}$.

3b.1) if $l^i_{RAT0} > 0$ broadcast an UPDT message indicating $|AT^i|$ and $D_i$'s id.

3b.2) if $l^i_{RAT0} < 0$ then determine how many devices, $n_d$, should take over the extra activity time consumed by device $D_i$ and broadcast an UPDT message with a Remote Activity Time Usage (RATU) flag indicating $|AT^i|$, $D_i$, $|l^i_{RAT0}|$, $n_d$ and a list of device's id. If $lastl^i_{RAT0} < 0$ then $|AT^i|$ is replicated in the $|l^i_{RAT0}|$ field as $D_i$ had already consumed all its local activity time. For the selected devices $j$, the LR-BS updates their $l^j_{RAT0}$ (stored in the table) accordingly, $l^j_{RAT0} = l^j_{RAT0} - |l^i_{RAT0}|/n_d$, and set $lastl^j_{RAT0} = l^j_{RAT0}$.

3b.3) save the current value of $l^i_{RAT0}$ into $lastl^i_{RAT0}$.

4) **Device $D_j$ receiving an UPDT w/RATU from LR-BS**

4a) if $D_j$ is in the list of devices, take the advertised $|l^i_{RAT0}|$ and update $l^j_{TAT} = l^j_{TAT} + |l^i_{RAT0}|/n_d$, $l^j_{RAT} = l^j_{RAT} - |l^i_{RAT0}|/n_d$ and $G^j_{AT} = G^j_{AT} - |AT^i| + |l^i_{RAT0}|$ because all $D_j$ in list of devices contribute to $|l^i_{RAT0}|$. Fig. 10.

4b) if $D_{j\neq i}$ is not in the list of devices update $G^j_{AT} = G^j_{AT} - |AT^i|$ because they have to remove what has been consumed by $D_i$. Fig. 10.

5) **Device $D_j$ receiving an UPDT from LR-BS**

5a) if $j \neq i$ then $D_j$ updates $G^j_{AT} = G^j_{AT} - |AT^i|$.

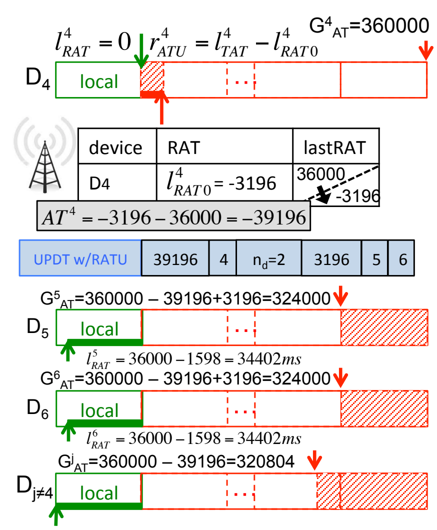


Fig. 10. Update of device's local Remaining Activity Time

In Fig. 10, let us take device $D_4$ as an example. $D_4$ used all his allowed local activity time, i.e. 36000ms, and also used $r^4_{ATU} = 3196ms$ from the remote activity time pool. Therefore $AT^4 = -39196ms$. When the LR-BS received the last image packet from $D_4$, it will decide, in the example, to assign to devices $D_5$ and $D_6$ the role of supporting the extra activity time consumed by $D_4$. Therefore the UPDT message with the RATU flag starts with the value of $|AT^4| = 39196ms$ followed by $D_4$'s id, $n_d = 2$, $|l^4_{RAT0}| = 3196$ from the table and finally $D_5$ and $D_6$ ids. $D_5$ and $D_6$ will each remove 1598ms from $l^5_{RAT}$ and $l^6_{RAT}$ respectively. If we assume that both devices did not send any message, then $l^5_{RAT} = l^6_{RAT} = 34402ms$. They then update their local value of $G^{5,6}_{AT}$ by removing $AT^4$, but adding $|l^4_{RAT0}|$ because both of them already contributed previously to $|l^4_{RAT0}|$. Therefore, at the end, they both have their $G^{5,6}_{AT}$ decreased by $D_4$'s whole duty-cycle. A device $D_{j \neq i}$ and not in the device list has to remove from its local value of $G^j_{AT}$ the totality of what has been consumed to have a consistent view for $G_{AT}$. As can be seen in Fig. 10, for any device $D_k$ the green arrow ($l^k_{RAT}$) and the red arrow (local $G^k_{AT}$) delimit the amount of total allowed activity time for that device.

### B. Example of a whole sequence with image transmission

Fig. 11 shows a whole sequence with an image transmission from device $D_4$. We assume that LoRa mode 1 is used for maximum range which is somehow a worst-case scenario.

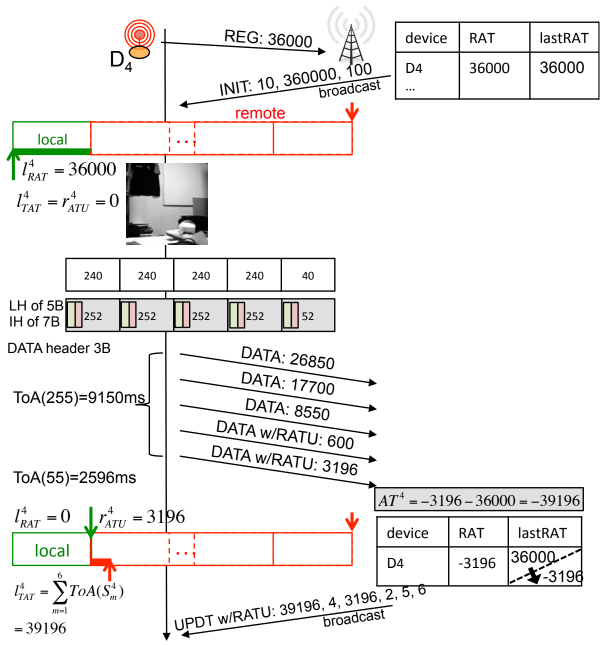


Fig. 11. Example of whole sequence with image transmission

After the initialization phase, $D_4$ knows the value of $G_{AT}$ and the number of devices, $n = 10$. We also assume that the LR-BS indicates devices could use 100% of the $G_{AT}$, i.e. $\alpha = 100$. Upon image change detection, $D_4$ encodes the image and we assume that 5 packets are produced: the encoded image size is taken on purpose to make the computation of ToA easy with Fig. 5. IMSS is set to 240B and the last packet is 40B long. Adding the Libelium header and the image header results in increasing the size of each packet by 12B. As the DATA packet needs 3 additional bytes, the ToA for LoRa mode 1 is computed on 255B except for the last packet where the final size is 55B. Each packet transmission of size $S^4_k$ will remove $ToA(S^4_k)$ from $l^4_{RAT}$. The transmission of the fourth packet consumes extra activity time and at the end $r^4_{ATU} = 3196ms$ have been consumed. At the end of the image transmission, the LR-BS broadcast an UPDT message as explained previously (see Fig. 10). If $D_4$ needs to send another image, it can continue consuming remote activity time.

### C. Packet format

We describe in Fig. 12 the packet format of our proposed activity time sharing protocol.

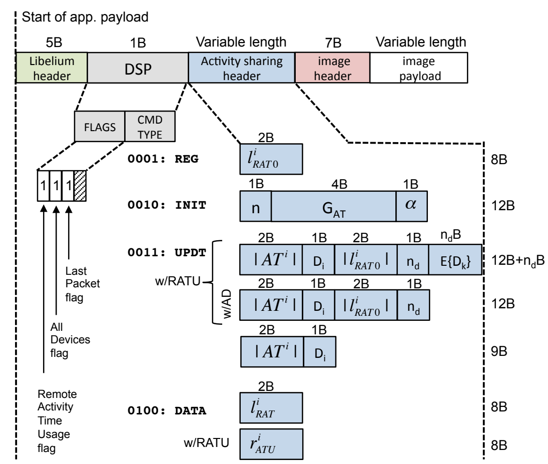


Fig. 12. Packet format

The first byte, DSP, contains two 4-bit fields for flag indicators and command type value. The RATU flag can be set to indicate whether the DATA packet carries an $l^i_{RAT}$ or an $r^i_{ATU}$ coded in the first 2 bytes after the DSP byte. The LP flag indicates that it is the last packet for the current image transmission. This flag can be set at the sender to better determine when the LR-BS can build UPDT messages. However, as the LR-BS also uses a timeout for each device, this flag is not mandatory. A DATA packet needs 3B plus 5B for the Libelium LoRa header, and plus 7B for the image header. Therefore the total DATA packet size with an IMSS of 240B is 255B which is the limit of the SX1272 radio in variable packet size mode. For the UPDT message, if the number of selected devices is greater than 243 (as the UPDT message needs 7B plus 5B for the Libelium header and 1B per device id, only 243 device id can be indicated in a single UPDT message), then another UDPT message can be sent for the remaining selected devices. However, the LR-BS can select all devices except the one that sent the image by using the All Devices flag (w/AD) to indicate that all devices $j \neq i$ should process the UPDT message with either action 4 or 5.

### D. Increasing robustness to packet losses

In the described mechanism, the LR-BS keeps track of consumed activity time when receiving a packet from a device $D_i$. However, it may happen that a packet sent by $D_i$ is not received by the LR-BS, for some reasons, while the activity time of this unsuccessful transmission should be counted. To make the system more robust to packet losses, the image data packet header includes for device $D_i$ the value of either $l^i_{RAT}$ or $r^i_{ATU}$, depending on the RATU flag in the packet header. In this case, actions 2c and 2d can be extended by respectively indicating $l^i_{RAT}$ or $r^i_{ATU}$ in the data packet header. Then action 3a can include a comparison between the $l^i_{RAT}$ indicated in the packet and the $l^i_{RAT0}$ stored by the LS-BR. A different value means that there have been some packet losses. Our implementation does this checking and the received $l^i_{RAT}$ replaces the $l^i_{RAT0}$ stored by the LS-BR.

### *E. Selecting devices to support consumed extra activity time*

In the example described above, action 3b.2 selected devices $D_5$ and $D_6$ to support the remote activity time consumed by $D_4$. In general, the LR-BS should avoid removing for each device $j$ an activity time greater than $l^j_{RAT}$. In this paper, we are not evaluating nor proposing a particular selection mechanism since our current implementation simply distributes the consumed activity time over all the other devices (use if the "All Devices" flag) therefore using the largest number of devices to reduce the amount of activity time penalization for a single device. However if it is desirable to have priority or exclusion mechanisms some particular devices can be selected and in this case it makes sense to selected those with the highest $l^j_{RAT0}$ and take into account a history record in order to avoid soliciting recently selected devices.

### *F. IFS for packet transmission*

Even if we are not using LBT+AFA to avoid the 1% duty-cycle, we propose to use LBT similarly to a carrier sense mechanism to improve reliability. LBT will be used in conjunction of a priority mechanism similar to the inter-frame spacing (IFS) mechanism of IEEE 802.11: Distributed IFS ($DIFS$) and Short IFS ($IFS$) where $SIFS < DIFS$. Both will be expressed in symbol period. We however propose a simplified backoff mechanism without exponential increase nor frozing&restarting the backoff timer as in WiFi networks. The LBT mechanism is based on the Channel Activity Detection (CAD) feature offered by the SX1272 LoRa chip.

Prior to send a DATA packet, an end-device should see a free channel for at least a $DIFS$ (we will refer to this case as a $DIFS_{CAD}$). If it is the case the packet is transmitted otherwise the device waits for a random number of $DIFS$ without performing CAD. At the end of the waiting period, the device will try again to have a $DIFS_{CAD}$. This process is repeated until the packet can be transmitted. However, to decrease the probability of concurrent image transmission, while the first DATA packet of an image enforces an $DIFS_{CAD}$, all following packets of the same image will only need an $SIFS_{CAD}$. In addition to reduce packet collisions and activity time wastage, having sequential image transmission facilitates actions 3b and 4 in order to have a consistent view for the various usage of remote activity time. All control messages sent by the LR-BS use the $SIFS$: INIT and UPDT messages will therefore have higher priority. For the specific case of REG messages that start every cycle, we propose to wait for a random number of $SIFS$ before performing an $SIFS_{CAD}$.

### *G. Cost of protocol message exchanges*

As shown in Fig. 12, the size of a REG message is 8B. If we assume LoRa mode 1 for maximum range, the ToA is 1122ms which should be taken from $l^i_{RAT0}$ in the REG message. In DATA packets, the additional cost of the activity time sharing mechanism (3B) can results in a ToA increase of 163ms, or zero, depending on the rounding effect on the payload size. The LR-BS is also constrained by the activity time regulation. The size of the INIT message is 12B and ToA is 1286ms. An UPDT message can use a large portion of ToA if the number of selected devices is large: selecting 6 devices gives a size of 18B and a ToA of 1449ms. For large networks, using the All Devices flag can limit the UPDT to 12B. The cost of action 3b.1 can also be made smaller by only triggering an UPDT message if $l^i_{RAT0}$ becomes small. If $l^i_{RAT0}$ is still large the system can easily run until the next update. Note that devices may wake up periodically to listen for potential UPDT messages that need to be transmitted at specific moment to share common activity periods.

## V. Conclusions

Visual surveillance applications with low-cost image sensors can become reality using long-range technologies to avoid complex and costly relay nodes. The challenge is to support the larger amount of data produced by image sensors while staying within the radio regulations defined for sub-GHz transmissions. We proposed an activity time sharing mechanism in scenarios where a pool of image sensors are deployed by a single organization: the activity time of all deployed devices are managed in a shared manner, allowing a device to transmit beyonds the 1% duty-cycle limit. The proposition is implemented on our low-cost image sensor platform and preliminary tests show that it is fully functional.

## References


[1] LoRa Alliance, "LoRaWAN™ specification, v1.0," 2015.
[2] Libelium, "Waspmote lora 868mhz 915mhz sx1272 networking guide. v4.0-11/2014," 2014.
[3] ——, "Extreme range links: Lora 868/915mhz sx1272 lora module for arduino, raspberry pi and intel galileo. http://www.cooking-hacks.com".
[4] ETSI, EN300-220-1. 2012.
[5] P. Chen et al., "Citric: A low-bandwidth wireless camera network platform," in *ACM/IEEE ICDSC*, Sept 2008.
[6] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin, "Camera mote with a high-performance parallel processor for real-time frame-based video processing," in *IEEE AVSS*, Sept 2007.
[7] Á. Rodríguez-Vázquez et al., "The Eye-RIS cmos vision system," in *Analog Circuit Design*, Springer Netherlands, 2008.
[8] W.-C. Feng et al., "Panoptes: Scalable low-power video sensor networking technologies," in *ACM TOMCCAP*, vol. 1(2), May 2005.
[9] Evidence Embedding Technology, "Cmucam: open source programmable embedded color vision sensors. http://www.cmucam.org/".
[10] Semtech, "Sx1272/73-860 mhz to 1020 mhz low power long range transceiver. rev.2-07/2014," 2014.
[11] C. Pham and V. Lecuire, "Building low-cost wireless image sensor networks: from single camera to multi-camera system," in *ACM ICDSC*, Sep 2015.
[12] V. Lecuire, L. Makkaoui, and J.-M. Moureaux, "Fast zonal dct for energy conservation in wireless image sensor networks," *Electronics Letters*, vol. 48, no. 2, 2012.