Edge-Computing Perspectives with Reconfigurable Hardware

Pascal Benoit, Loïc Dalmasso, Guillaume Patrigeon, Thierry Gil, Florent Bruguier, Lionel Torres

Edge-Computing Perspectives with Reconfigurable Hardware

Pascal Benoit, Loic Dalmasso, Guillaume Patrigeon, Thierry Gil, Florent Bruguier, Lionel Torres
LIRMM – CNRS UMR 5506,
University of Montpellier
161 rue Ada, 34095 Montpellier, France

Abstract— **The Internet of Things is a promise of smarter technologies, with devices working together in a distributed manner, to provide more quality of service in many domains, such as industry, transports, energy, health, etc. Fog/Edge computing is probably one of the most interesting concepts as it will be a means to optimize energy and performance. However, beyond the principle, few works have really demonstrated the real potential of it, as many challenges need to be addressed at different levels: hardware design, software APIs, communication protocols, radio technologies, and so on. In this context, there is a growing interest in reconfigurable devices, as they bridge the gap between performance and energy with the advantage of the hardware flexibility. In this paper, we present a platform with sensor nodes and gateways relying on FPGAs, and showing that reconfigurable devices can be an enabling technology for edge-computing.**

Keywords—Internet of Things; Fog computing; Gateway; Data Distribution Service; M2M; FPGA

I. Introduction

The Internet of Things is a concept that extended 10 years ago the one of wireless sensor networks, where each node is a device able to interact with its environment, as a sensor or an actuator, and connected directly or indirectly to the Internet over TCP/IP. A further extent of the concept talks about Internet of Everything, which means that in the future, anything in this world, will be equipped with a piece of electronic that will connect it to the global network. Machine to machine (M2M) communication is strongly bound to IoT, which means that the devices are able to send/receive data and interact together autonomously. M2M and IoT can bring revolutionizing applications in many domains, such as the industry (process monitoring, predictive maintenance), transports (traffic optimization, emergency management), energy (production and consumption balance, resource optimization), health (patient monitoring), and many others.

IoT is a promise for about ten years now, and despite many contributions and huge efforts in the research and industry communities, there are still big challenges to address before turning it into a reality. One of the reasons is that behind the concept that is very easy to understand, implementing the IoT rises many problems: interoperability, security, data storage management, acceptability, privacy, wireless networks maturity, scalability, energy, quality of service, etc., as highlighted in [1]. Today, there are successful demonstrations of the concept, but mostly limited to a very specific application, in a given context, with customized devices and protocols. But according to CISCO in 2017, 75% of IoT projects failed before completion, which shows that it is mandatory to build a strong scientific body to study, understand, design and optimize the Internet of Things, so that it fulfils its commitments. Many fundamental questions arise from a scientific perspective in the context of IoT: what is the “object” of study, what are the building blocks, how to model globally, how to model accurately, how to segment, how to compare centralized vs. distributed approaches, specific vs. generic, how to optimize the energy, how to address the security, how to measure the Quality of Service of a given application, etc.? Eventually, can IoT be apprehended as a whole, as it would be possibly the connection of everything?

Our ambition is more modest but is an important milestone in addressing a number of the issues raised above: the objective is to provide a complete prototyping platform with a set of flexible and customizable hardware and software, in order to explore innovative approaches in the context of IoT, such as Fog computing, where storing and processing capabilities are decentralized to the edges of the network e.g. nodes or gateways. The motivation is to rationalize data exchanges, endowing processing capabilities into edge devices or even sink nodes, allowing only meaningful information to be communicated. This approach could enable near-sensor analytics, as the processing of data would be performed close to the source of information. There is clearly a growing interest in this concept, as evidenced by recent research papers [2][3][4]. However, devices like sensor nodes or gateways required to support this new paradigm, do not provide the necessary hardware, software and communication protocols to implement it efficiently, as their role is generally limited: sense/send/receive data for the nodes, or bridge the gap between sensors and an IP network for the gateways.

We have developed an IoT platform composed of a set of reconfigurable sensors and gateways, which provides all the building blocks required to explore, design and implement the concept of Fog Computing. FlexNode is a prototyping board using an FPGA for design space exploration at the sensor level: the architecture relies on a generic MCU that can be customized on purpose, *i.e.* processor, dedicated blocks, accelerators, memory, etc. Reconfigurable Gateways are built on a Zynq FPGA, integrating a Processing System (with a dual core ARM) and Programmable Logic, that can be configured to offload time or power-consuming tasks. Gateways are designed to support M2M communications, to store and to process data in a distributed manner, allowing potentially more performance, reliability and scalability. Their design is generic: both software and hardware are flexible and customizable, so that they can be adapted to various requirements, easing the same way interoperability and reducing the costs. In this paper, we present the building blocks of the platform, and demonstrate with different use cases the advantages of reconfigurable hardware to explore the concept of Fog Computing.

The remainder of this article is organized as follows: part II is dedicated to related works; then we provide a description of the FlexNode in the third part. Section IV summarizes the characteristics of the gateway, hardware and software elements, and the communication protocol used to enable M2M interactions; case studies are reported in the next section, to demonstrate the relevance of such an approach, with some results showing the advantages provided by reconfigurable computing in the Fog.

II. Related works

The utilization of FPGAs in sensor nodes has been investigated for several years. One approach is to use it directly in replacement or additionally to an MCU. Recently, [5] have shown a Zynq FPGA platform used to collect tri-axes vibration data and performs FFT computations in a high performance wireless sensor node. In [6] a Virtex-4 FPGA is used to design a wireless sensor node, with runtime configuration capabilities. An underwater acoustic module based on FPGA was demonstrated in [7]. The FPGA can also be used for specific tasks, like security services at the sensor level, as in [8] where Hyperchaos Encryption is implemented in the sensor node within the FPGA. Reconfigurable devices have demonstrated strong abilities to perform signal and image processing, and are particularly well suited for vision sensor nodes, as it is shown in [9], where they integrate a Nios II soft-core processor, image acquisition and compression circuit on a single FPGA chip, which meets the design requirements of low-power consumption, flexibility, low-cost and in small size. A Zigbee image sensor node is presented in [10] where a tracking application is demonstrated with 2 sensor nodes. The advantage of dynamic reconfiguration in the sensor node to reduce redundant transmission, consumes less power and bandwidth in the context of a surveillance system was reported in [11]. Another approach is to use the FPGA as an emulator or as a prototyping platform for hardware/software evaluations. In [12], the prototyping boards based on an FPGA are used for fast implementation and verification. SUNSHINE [13] is a hardware-software emulator allowing the simulation of flexible sensor nodes.

Gateways have a key role in the context of IoT: they implement the networking protocols, distribute the storage resources, allow potentially edge analytics, and secure data from things to the cloud. In [2], authors discuss the need of extra functionality in gateways to perform processing on data before sending to the Internet: they clearly show that adding some intelligence at the edge would enable a better utilization of network resources, and improve the performance of applications. In [14], it is shown that efficient architectures are missing to provide access to the Internet for low-power devices: for this purpose, smartphones are suggested to support the gateway functionalities.

Commercial dedicated gateways instance from Advantech, Multitech, Huawei, Dell, HPE, etc., feature radio (BLE, WiFi, GSM, 4G-LTE, 3G, LoRa, etc.) to Internet hardware and software resources, generally based on general purpose approaches like embedded PCs (ARM-based or Intel-based architectures) running Windows (10 IoT enterprise) or Linux (Ubuntu Snappy, Windriver). But it was shown that traditional software-based gateways are limited in terms of performance. The authors of [15] show that a Xilinx FPGA gateway system was able to reduce up to 94.7% on execution time compared to related works. In [16], they propose to use an FPGA to perform protocol conversion and secure transmission between 4G and PROFIBUS-DP. In [17], the authors present a configurable vehicular Ethernet gateway utilizing a hybrid FPGA, with interesting capabilities like run-time adaptability of the switch to address network security in connected vehicles. Reconfigurable gateways were also used in the context of Intelligent Home: [18] reports a hybrid ARM+FPGA system, which realizes the connection between home appliances and the Internet over ZigBee and Wi-Fi networks. A similar contribution [19] was sown over a cellular network. Security is also a big motivation of reconfigurable gateways, as highlighted in [20] to implement a VPN and secure data transfers over non-confidential network areas.

Reconfigurable hardware is considered now for several years as a relevant approach in the context of WSN and IoT, and is gaining more and more interest both at the sensor and gateway levels, as demonstrated in this related works section. However, it can be noticed that the computations and tasks handled by reconfigurable devices do not really show the exploration or implementation of a Fog Computing approach, *i.e.* the ability of the building blocks to distribute computation / storage and communicate over the network in a collaborative manner.

III. FlexNode architecture

The FlexNode prototyping platform consists of an elec- tronic board with a controller slot, peripheral slots, power distribution and the necessary components to perform power characterization of each element composing the node. The peripheral slots can be used to connect sensors, actuators or communication modules.

This platform can be used primarily for the evaluation and comparison of multiple controller solutions, with different node architectures, peripherals, sensors, radios and applications. This node can be used directly as a WSN to perform evaluation while taking account of communication and network hazards.

The Digilent CmodA7 was chosen as the central unit thanks to its small size, 48-pin DIP board built around a Xilinx Artix-7 FPGA, the XC7A35T-1CPG236C, which has 20 800 LUTs, 41 600 Flip- Flops, 225 kB of RAM and 1 MSPS ADC. It is possible to explore different architecture solutions by reproducing the desired behavior within the FPGA.

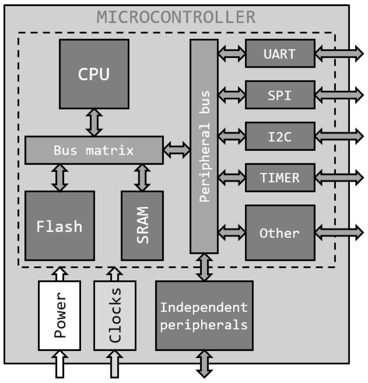


Figure 1 : Generic architecture of FlexNode

Typical microcontrollers include at least one processor, a non-volatile memory (usually Flash for code instructions and read-only data), a volatile memory (usually SRAM for application data), a power management unit, a clock management unit, input/output peripherals, communication modules (UART, SPI, I2C, USB, CAN. . .) and timers. This typical architecture is depicted in Fig. 1. Some microcontrollers also include different types of non-volatile memories (ROM, EEPROM...) or have a multi-master system (multi-processors, Direct Memory Access (DMA)...). Depending on the application, this architecture can be customized, which implies a potential large design space to explore.

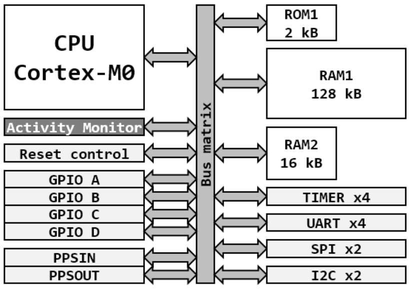


Figure 2 : Customized Flexnode

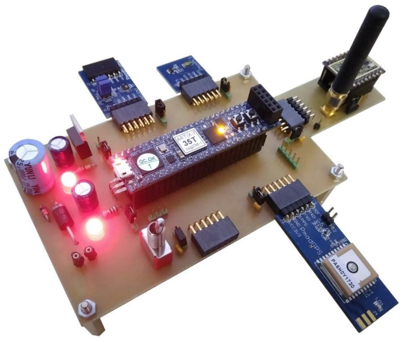


Figure 3 : One instance of FlexNode with a LoRa radio

As a demonstration example (Fig. 2), we use the ARM Cortex-M0 r1p0. This is a 3-stage 32-bit RISC processor that implements the ARMv6-M ISA. It includes a single AHB-Lite interface, 32 interrupt lines, 1 Non-Maskable Interrupt and a single-cycle multiplier. The architecture also includes a 2 kB ROM containing a bootloader code, a 128 kB RAM and a 16 kB RAM, peripherals for inputs/outputs control (44 I/O, PPS), serial communication (4x UART, 2x SPI, 2x I2C) and timing modules (4x 16-bit timers). All these elements are connected together thank to a single-master AMBA3 AHB-Lite system. A peripheral called Activity Monitor is also used to report events and characterize power consumption and performance.

The complete board is depicted in Fig. 3. We use the ADT7420, a digital temperature sensor with I2C interface, which is the one used by Digilent on the PmodTMP2. The radio module is the SX1272 from SEMTECH, a LoRa transceiver with SPI interface. The SX1272 is configured in LoRa mode with a spreading factor of 12. This system has also been validated with the Pmod BLE (Bluetooth Low Energy), Pmod GPS receiver, Pmod ALS (Ambient Light Sensor) and Pmod HYGRO (Digital Humidity and Temperature sensors) modules from Digilent.

FlexNode provides therefore a generic hardware architecture that can be customized, implemented in an FPGA hosted on a modular board. It is possible to explore the design of the MCU architecture (processor complexity, pipeline stages, custom computing units, memory architecture, memory size, peripherals, …) directly in the context of a WSN, *i.e.* taking into account the specificities of the application, sensors/actuators, radio environment. It enables the assessment in a real context, to measure/compare performance and energy, make design choices to choose an existing platform that fits the best tradeoff. Furthermore, the exploration includes the possibility to perform custom operations at the Sensor Node level, *i.e.* implement data processing (*e.g.* statistics), signal processing (*e.g.* filtering), before transmitting data, including thus edge computing capabilities at the sensor level thanks to the reconfigurable hardware.

IV. RECONFIGURABLE HYBRID GATEWAY

Nodes may potentially communicate together, through direct point to point connections, but in order to widen interaction capabilities and services, giving access to the IP network is a great opportunity. It is the first role of the gateway, to connect “radio world” to the “TCP/IP world”: for this purpose, we present here a generic gateway architecture allowing M2M communications. It brings all the software and hardware building blocks to decentralize processing to the edges of the network. In [21], we were introducing this project combining local processing, with completely decentralized communications and configurability, so that a network can cooperate in the most efficient way. As a point of comparison, the Open IoT gateway in [22] is very configurable, but operates on a Cloud-based network, and the Smart Gateway presented in [23] uses Message Queuing Telemetry Transport as a communication protocol, while we chose Data Distribution Service as the communication protocol.

The gateway is based on a Zybo board from Digilent (Figure 4). This board integrates a Xilinx Zynq chip, which is a System on Chip composed of a processing system (PS) and programmable logic (PL). The PS is based on an ARM Cortex-A9 dual-core, which runs at 650MHz. The board has several interfaces: USB, HDMI, VGA, Ethernet, audio, as well as six 12-pin ports called Pmods. Some of the interfaces are directly linked to the ARM processor, while others need to be managed in hardware.

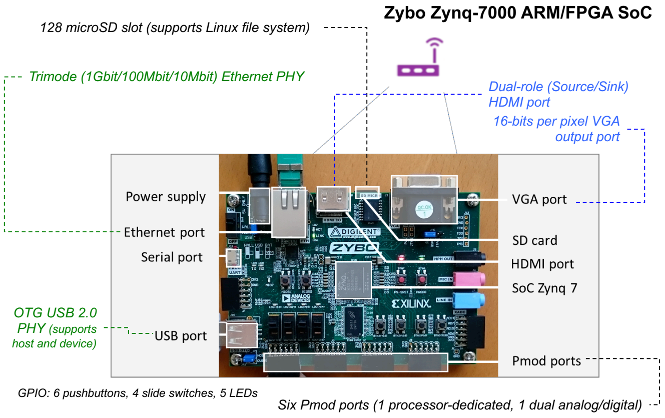

Zybo Zynq-7000 ARM/FPGA SoC
128 microSD slot (supports Linux file system)
Trimode (1Gbit/100Mbit/10Mbit) Ethernet PHY
Dual-role (Source/Sink) HDMI port
16-bits per pixel VGA output port
Power supply
Ethernet port
Serial port
USB port
OTG USB 2.0 PHY (supports host and device)
VGA port
SD card
HDMI port
SoC Zynq 7
Pmod ports
GPIO: 6 pushbuttons, 4 slide switches, 5 LEDs
Six Pmod ports (1 processor-dedicated, 1 dual analog/digital)

Figure 4: The Gateway based on Zybo board

The USB port and the different Pmod ports can be used to attach controllers to manage sensors through different radios. The gateway supports currently Bluetooth Low Energy, ZWAVE and LoRa. Since the PL is reconfigurable hardware, it can be customized on purpose, for hardware acceleration, including statistical analysis, cryptographic primitives, video processing, etc. The idea is to allow locally extracting information from heavy data (a video stream for instance), thus reducing the bandwidth required. PL is also important to keep the gateway generic: it can be adapted for many use cases. The SoC is built in a way that makes the exchange of information between PS and PL efficient.

On top of the Linux system based on Yocto Project [24], the software architecture that is running on the gateway is depicted in fig. 5. This architecture is composed of processes that control different peripherals (Bluetooth, Zigbee, ZWAVE, LoRa, etc.), enabling sensor data to be collected. In terms of communication protocol, in the context of IoT especially for edge-computing purposes, the publish-subscribe pattern is more appropriate than server-client. It has been demonstrated in [21] that Data Distribution Service (DDS) outperforms Message Queuing Telemetry Transport (MQTT) when comparing latencies, which emphasizes its real-time capabilities, despite that bandwidth requirements are larger for DDS. However, the fact that data are potentially processed locally in the gateway decreases bandwidth needs. An open-source solution has been chosen, namely OpenDDS, in order to keep our project generic [25]. A DDS topic consists in a structure of several data fields. These topics are made by the user, so they are entirely application-specific. For instance, a temperature topic can be created to transport the temperature, the node ID and a timestamp. This is how gateways communicate with one another. The *processing unit* is responsible of multiple tasks. It can process data to extract useful information, store in a database, and transmit data to the *sender* (these data will have been pre-computed). Finally, the *receiver* is the process that reads data from the TCP/IP network and that forwards them to the processing unit. These data come from other gateways in the network. They can be for example data from distant sensors that can be useful for the local gateway. In order to store data in a generic, compact and flexible manner, the SQLite database [26] is used to build a relational database. The entire database is stored in only one file, and can be managed either by issuing requests in command-line or directly in a C code.

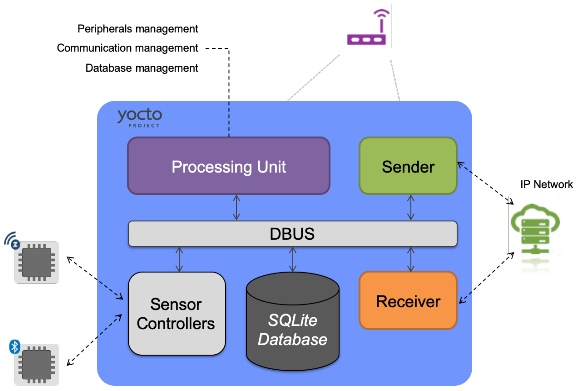


Figure 5: Software architecture of the Gateway

As FlexNode, the Reconfigurable Gateway provides a generic hardware architecture that can be easily customized. Based on a SoC FPGA hosted on a modular board, it is possible to generate a Linux distribution adapted to the implemented hardware, to customize software in the PS and the design in the PL, adding for instance dedicated interfaces, or accelerators. It enables gateway prototyping in a real context to assess performances on a given application with real wireless sensors. The reconfigurable logic is a strong advantage in the context of IoT, as it gives the possibility to implement complex processing at the Gateway level.

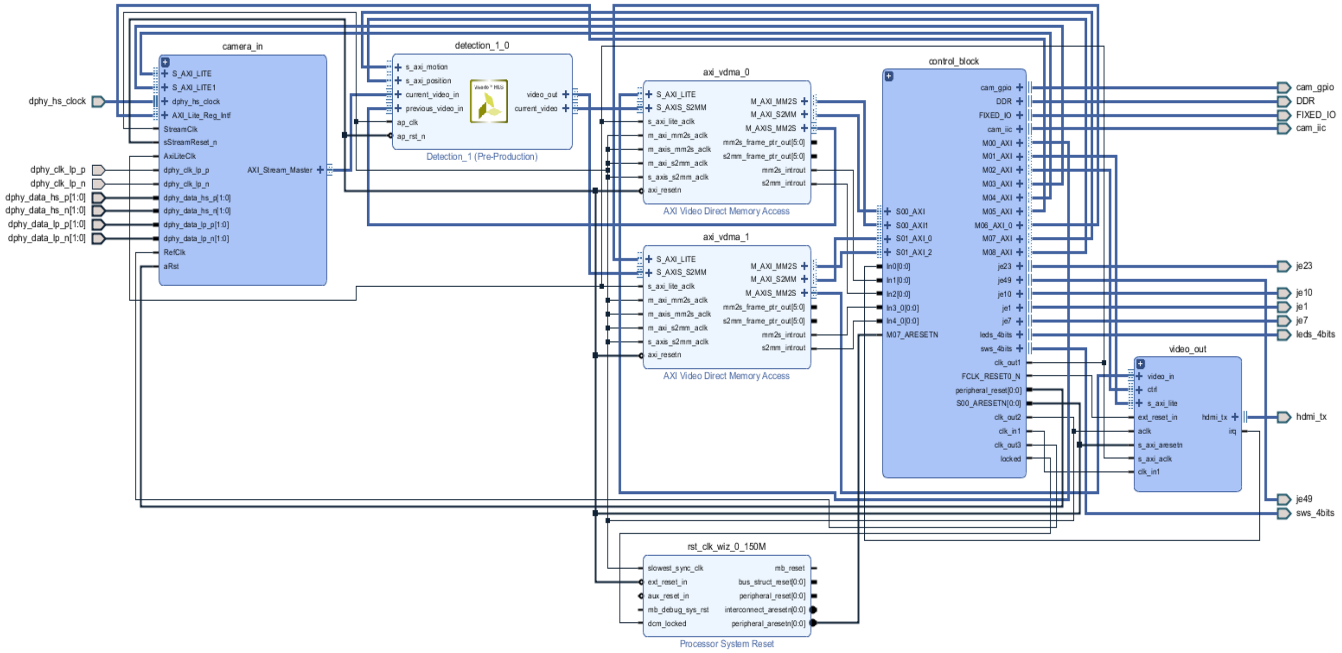


Figure 7: Block design of the smart camera architecture

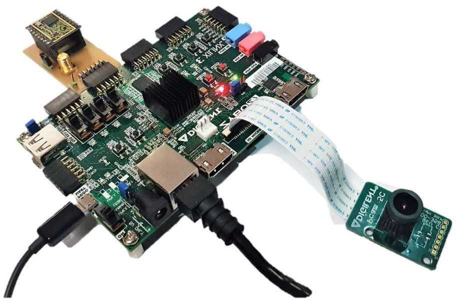


Figure 6: Hybrid Gateway

As an example, we explored the possibility to use the design of the reconfigurable gateway as a Smart Camera. The prototype called "Hybrid Gateway" is depicted on figure 6. This system running a custom Linux distribution includes a LoRa pmod module (designed in our lab), an Ethernet connection, and a Pcam 5C (5 MP Fixed Focus Color Camera Module), connected to the ZYBO Z7 board (based on a Zynq Z-7020) with a Pcam MIPI CSI-2 connector. A simple motion detection accelerator was designed with VIVADO HLS. This hardware block, which is mapped in the PL and communicates with PS through AXI VDMA, handles the video input stream from the camera, the video output stream (optional) and the detection block that compares successive images to identify right to left, or left to right motions. The block design in VIVADO is depicted on figure 7. Processing the video stream on board enables opportunities like communicating visual information through bandwidth limited networks like LoRa, or simply drastically reduce the bandwidth utilization on a TCP/IP network. This is also an interesting means to avoid the transport of private/sensitive data over different networks, by extracting only the necessary required information.

V. Case Studies

The first purpose of the Flexnode is prototyping, performance and power consumption assessment. The board was designed in order to measure easily the power consumed by each subsystem of the sensor node (FPGA board, sensors, radios, etc.). Four external peripherals were used in this case study: Pmod ALS, Pmod HYGRO, Pmod LoRa, Pmod GPS. In figure 8, we observe 4 periods of 5 seconds each (SENSE phase, PROCESS phase, SEND phase). There is a peak power of 650 mW while during SLEEP mode the average consumption is 570mW (obviously we are far from the numbers of general purpose MCUs, but these numbers can be considered relatively for comparison purposes). The consumption of the sensors is relatively low (≈130mW) compared to the one of the CMOD A7 (≈470mW). The architecture inside the FPGA is instrumented so that we are able to relate logical events obtained with performance counters, to the power that is consumed, and estimate the power that would be consumed on a given technology node. More details on the methodology can be found in [27]

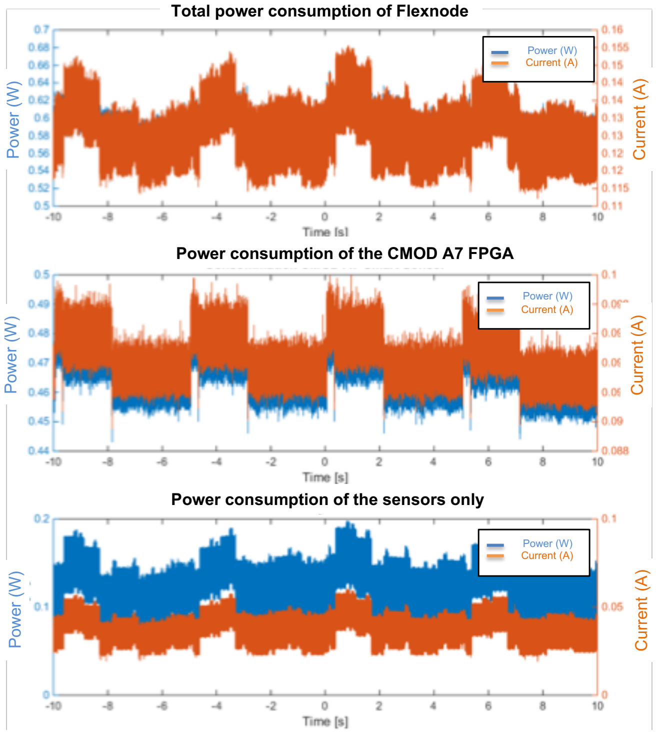


Figure 8: Power consumption evaluations in Flexnode

A prototype of the whole network has been then developed for demonstration, which uses three sensor networks: Bluetooth Low Energy, LoRa and Z-Wave (proprietary protocol developed by Sigma Designs working on the 868MHz ISM band in Europe). The main claim is to demonstrate the utilization of reconfigurable hardware in the context of IoT, to process and communicate data in a decentralized network, from different radios over a TCP/IP network. The prototype, shown in fig. 9, validates and demonstrates the potential of the network of sensors and gateways in a smart office application. The gateways collect information from the three sensor networks, and publish them on different DDS topics. The information shared over the network are: temperature, humidity, ambient light, door sensors, presence sensors and video-related data (motions). For each one, a topic is created; *publishers* and *subscribers* are manually configured. Each sensed data is timestamped and geo-localized when available.

Another interesting approach in IoT is the capability to access and aggregate data from different sources. In order to study this aspect, we developed special programs on the gateways that are able to collect information from external sources. For demonstration purposes, we use the data available from the https://data.montpellier3m.fr/. It is possible to download *xml* files that are periodically updated, for instance the number of available parking spaces in each parking of the city. This opens the possibility of merging data from different sources in the gateway databases, which can be used to manage efficiently the network.

The shared information in the network can be visualized at publication time in the terminals of the gateways, or accessed in the local databases. In order to provide a user-friendly interface for visualization, we developed a website hosting Grafana, which is a platform allowing the monitoring of data. The PC running the website subscribes to all the topics in the network, and stores all the timestamped collected data in an *infuxDB* database. Then, it is possible to generate a custom dashboard with widgets allowing to visualize collected data (histograms, charts, etc.). Screenshots from Grafana are depicted in figure 10 and 11: they show data from a temperature sensor, a door sensor, a motion sensor, the number of available parking spaces in the “Comédie parking” in Montpellier, and its evolution from 10:00 AM to 4:00 PM the same day.

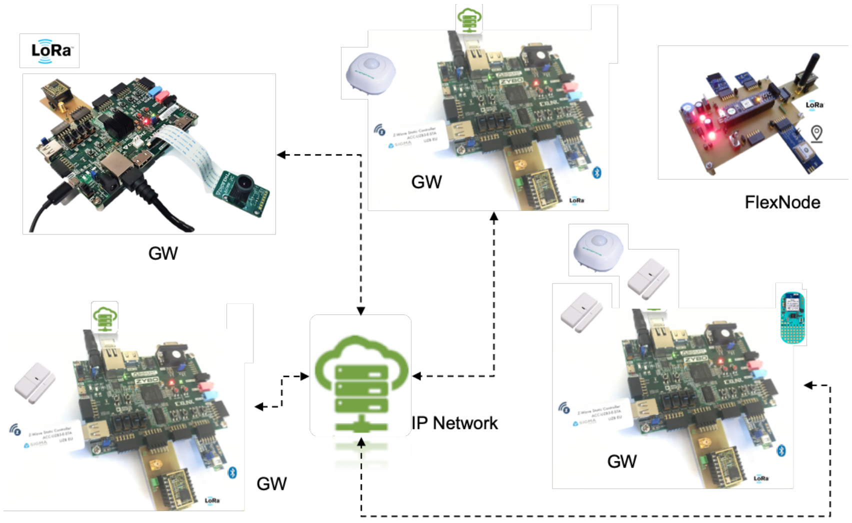


Figure 9: Demonstrator with 4 gateways and 8 sensors

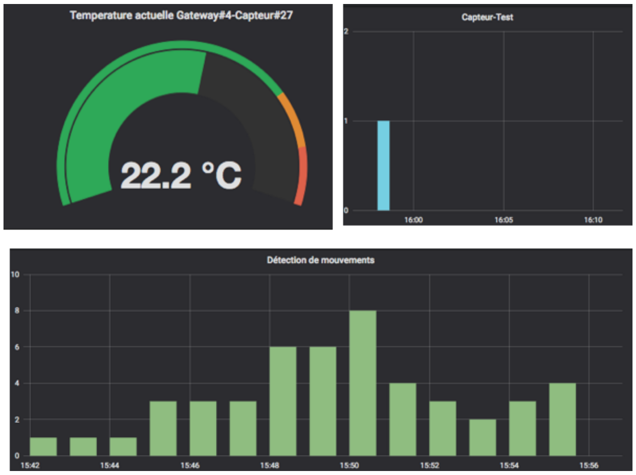


Figure 10: Visualization of internal data with Grafana

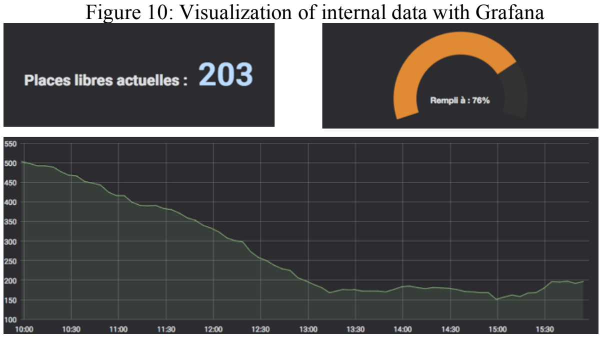


Figure 11: Visualization of external data with Grafana

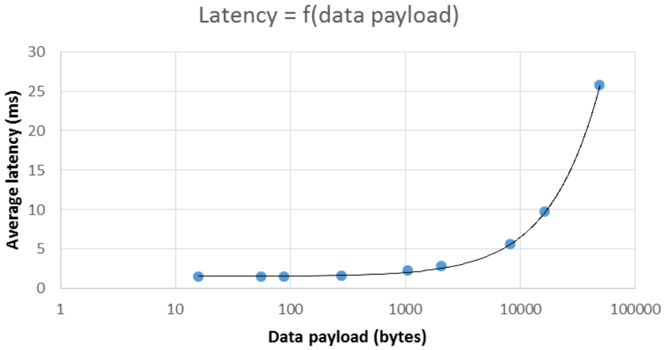
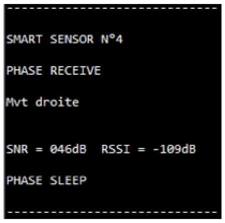


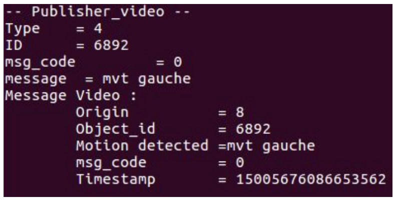
Figure 12: Sample latency compared to data payload

Several performance tests have been realized to show the capabilities of the developed network. A very important metric to measure performance is the latency, as it impacts the reactivity of the network for a given application. We made an experiment to characterize the latency, computed as the total time of the experience divided by the number of samples transmitted, with different payload sizes. DDS was configured to use the RTPS-UDP transport setting. For each data payload, between 10,000 and 100,000 samples have been used. In the figure 12, we can notice that for small payloads (less than 2 kB), sample latency remains approximately the same. However, after that point, sample latency begins to increase rapidly. At 8 kB, latency is doubled, and at 16 kB it is again doubled. The goal here is to observe latency in an absolute way. What this figure shows is that sending smaller payloads helps achieving better latencies. Decreasing the order of magnitude of data payload by processing them locally can bring a significant decrease in latency. The goal of local processing is to extract useful information to reduce the payload to its minimum. All information that would be deemed useless would not be transmitted. This introduces adaptive capabilities to the whole network that will be able to tune its behavior according to the context, which is a big requirement for IoT applications.



```
----------------------------
SMART SENSOR N°4

PHASE RECEIVE

Mvt droite


SNR = 046dB   RSSI = -109dB

PHASE SLEEP

----------------------------
```

Figure 13: Terminal view from the Flexnode showing that the node received an information of a left to right motion (Mvt droite) from the smart camera



```
-- Publisher_video --
Type     = 4
ID       = 6892
msg_code          = 0
message  = mvt gauche
Message Video :
         Origin           = 8
         Object_id        = 6892
         Motion detected =mvt gauche
         msg_code         = 0
         Timestamp        = 1500567086653562
```

Figure 14: Terminal view from the Gateway showing the publication on the “Video” topic of a right to left motion (Mvt gauche) event

Another test was realized with the smart camera, a Flexnode and the whole network of gateways. In this scenario, the smart camera is able to communicate information with LoRa to the LoRa Flexnode (Figure 13), and to publish information on the network about detected motions (Figure 14). This simple application illustrates the drastic reduction of required bandwidth: the initial video stream is about 250MB/s. It is in this case impossible to transmit this over a LoRa network, and it potentially consumes a lot of the available bandwidth in a TCP/IP network (25% if Gigabit Ethernet). After video processing, the data to be transmitted is only 1 Byte on event. It is a drastic reduction, but as mentioned above, small payloads are mandatory both for sparing bandwidth and reducing latency, as it impacts directly the reactivity of the network.

VI. Conclusion and Perspectives

We presented a complete platform with sensor nodes, gateways and smart cameras relying on FPGAs, and a visualization tool based on Grafana, showing that reconfigurable devices can be an enabling technology for edge-computing. They primarily provide a prototyping environment that can help exploring hardware and software at a very fine grain, allowing thus to define the best trade-offs for a given application that can be evaluated in a real context. Moreover, FPGAs offer local processing resources that can be customized, reconfigured and adapted, which opens great opportunities to respond to the challenges of current IoT networks by reducing the volume of information that transits through the IP network, thus enabling the promises of Edge Computing. We demonstrated that our approach has many advantages as it is fully distributed, allowing potentially more performance, reliability and scalability. It relies on technologies that can be adapted to various requirements, easing interoperability and costs reduction. Future works will be focused on developments including the design and integration of hardware IPs for applications in the fog.

VII. Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687973 - GREAT (heteroGeneous integRated magnetic tEchnology using multifunctional standardized sTack (MSS)) and the French National Research Agency under grant ANR-15-CE24-0033-01 (MASTA project).

VIII. References

[1] A. Hakiri, P. Berthou, A.Gokhale, and S.Abdellatif, "Publish/suscribe-enabled software defined networking for efficient and scalable IoT communications", *IEEE Commun. Mag.*, vol. 53, n° 9, pp. 48-54, sept 2015

[2] F. Bonomi "Fog Computing and its Role in the Internet of Things" Proc. Wksp. Mobile Cloud Computing pp. 13-16 2012-Aug.

[3] Mohammad Aazam, Pham Phuoc Hung, Eui-Nam Huh, "Smart Gateway Based Communication for Cloud of Things", *In the proceedings of 9th IEEE ISSNIP*, Singapore, 21-24 April, 2014

[4] Mugen Peng, Shi Yan, Kecheng Zhang, and Chonggang Wang. 2016. Fog-computing-based radio access networks: issues and challenges. *Netwrk. Mag. of Global Internetwkg.* 30, 4 (July 2016),46-53

[5] B. BILLEL, O. Z. MOHAMED, C. SAMIR, T. ABDELMOGHNI, M. MOHAMED and L. SIDAHMED, "High-performance Zynq FPGA-based wireless sensor node for vibration monitoring systems," *2018 International Conference on Applied Smart Systems (ICASS)*, Medea, Algeria, 2018, pp. 1-5

[6] S. Meena and N. K. Prakash, "Runtime reconfiguration of wireless sensor node using FPGA," Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, 2014, pp. 1-5

[7] Liu and S. Wang, "Design and realization of a QPSK modem module based on FPGA for underwater sensor node," Proceedings of the 29th Chinese Control Conference, Beijing, 2010, pp. 4781-4785

[8] J. Tong, Z. Zhang, Q. Sun and Z. Chen, "Design of Wireless Sensor Network Node with Hyperchaos Encryption Based on FPGA," 2009 International Workshop on Chaos-Fractals Theories and Applications, Shenyang, 2009, pp. 190-194

[9] C. H. Zhiyong, L. Y. Pan, Z. Zeng and M. Q. -. Meng, "A novel FPGA-based wireless vision sensor node," 2009 IEEE International Conference on Automation and Logistics, combine , 2009, pp. 841-846

[10] H. Kikuchi and K. Morioka, "Development of wireless image sensor nodes based on FPGA for human tracking in intelligent space," IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, 2012, pp. 5529-5534

[11] P. Latha and M. A. Bhagyaveni, "Reconfigurable FPGA based architecture for surveillance systems in WSN," 2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC), Chennai, 2010, pp. 1-6

[12] U. Quadri, P. Rangaree and G. M. Asutkar, "FPGA implementation of an emulator for Wireless Sensor Node with Pt100 temperature sensor," 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), Xi'an, 2013, pp. 1-5.

[13] J. Zhang, S. Iyer, P. Schaumont and Y. Yang, "A Simulator for Flexible Sensor Nodes in Wireless Networks," 2011 Seventhùù International Conference on Mobile Ad-hoc and Sensor Networks, Beijing, 2011, pp. 373-375

[14] Thomas Zachariah, Noah Klugman, Bradford Campbell, Joshua Adkins, Neal Jackson, and Prabal Dutta, « The Internet of Things Has a Gateway Problem », *In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 27-32, 2015

[15] T. Lee, C. Kuo and I. Lin, "High performance CAN/FlexRay gateway design for in-vehicle network," 2017 IEEE Conference on Dependable and Secure Computing, Taipei, 2017, pp. 240-242

[16] Y. Zhou, W. Xiao, M. Liu and X. Li, "Design of the embedded gateway for 4G and PROFIBUS-DP based on FPGA," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2017, pp. 748-752

[17] S. Shreejith et al., "VEGa: A High Performance Vehicular Ethernet Gateway on Hybrid FPGA," in IEEE Transactions on Computers, vol. 66, no. 10, pp. 1790-1803, 1 Oct. 2017

[18] W. Zhou, Z. Zhang, B. Zou and H. Nei, "Research of Intelligent Home Heterogeneous Gateway," 2013 International Conference on Computer Sciences and Applications, Wuhan, 2013, pp. 187-189

[19] W. Tsai, S. Zhu, M. Lu, J. Merzoug, C. Yu and I. Huang, "An implementation of IoT gateway for home appliances control over cellular network, »

[20] L. Shaofeng, G. Chaoping and S. Weifeng, "Design and Implementation of an Enhanced VPN Isolation Gateway," 2017 International Conference on Robots & Intelligent System (ICRIS), Huai'an, 2017, pp. 82-85

[21] J. Laurent, P. Benoit, L. Dalmasso and T. Gil, "Computing in the Fog with Reconfigurable Gateways," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, 2018, pp. 1-4

[22] Shiratech. "Open IoT Gateway". [Online] Available at : http://www.shiratech.com/open-iot-gateway/ [Accessed May 2019]

[23] Patrick Steiner, "Red Hat IoT demo – Next generation". 2016. [Online]. Available at : http://www.opensourcerers.org/red-hat-iot-demo-next-generation/ [Accessed May 2019]

[24] Linux Foundation. "Yocto Project". [Online]. Available at : https://www.yoctoproject.org/ [Accessed May 2019]

[25] Object Management Group. "OpenDDS". [Online]. Available at : http://opendds.org/ [Accessed May 2019]

[26] SQLite, [online] available at : https://www.sqlite.org/ [Accessed May 2019]

[27] G. Patrigeon, P. Leloup, P. Benoit and L. Torres, "FlexNode: a reconfigurable Internet of Things node for design evaluation," *2019 IEEE Sensors Applications Symposium (SAS)*, Sophia Antipolis, France, 2019, pp. 1-6