# Towards an FPGA-Based Edge Device for the Internet of Things

T. Gomes, S. Pinto, T. Gomes, A. Tavares, J. Cabral
Centro Algoritmi - University of Minho
{mr.gomes, sandro.pinto, tgomes, atavares, jcabral}@dei.uminho.pt

***Abstract*—With the growing ubiquity of Internet of Things (IoT), myriads of smart devices connect and share important information over the internet. In order to provide connectivity and interoperability of all the existing heterogeneous wireless devices, a full communication stack is proposed by the IoT Architecture Reference Model (IoT-ARM). From the sensor to the cloud, the proposed stack can be implemented on all IoT devices avoiding the battle for the wireless standard that will be adopted. This work in progress paper proposes an FPGA-based edge device for IoT, which uses SoC (System-on-Chip) FPGA technology to offload critical features of the communication stack to dedicated hardware, aiming to increase systems performance.**

***Index Terms*—Internet-of-Things (IoT), IoT-ARM, packet processors, SoC FPGA, 6LoWPAN.**

## I. Introduction

WSNs have became very popular on a wide range of domains, such as critical monitoring systems, security, health care or industrial applications [1,2]. The low power sensing nodes usually comprises a self-forming and self-healing mesh network, where a big number of devices communicate with each other, collecting data and send them to a centralized controlling application. The need to send the collected data over the internet to dedicated online services has considerably increased. This represents a big shift on the WSNs paradigm and introduces the new era of the Internet of Things.

IoT is already here and in use. Nowadays IoT represents a collection of billions of tiny smart connected devices and sending secured data through the internet to dedicated online cloud servers [3]. According to [4], it is expected by 2020 to exist around 30 billion of smart devices installed and connected to the cloud, representing four devices for each individual person in the world. Heterogeneous devices from different vendors, with unique and distinct hardware and software implementations, gather and send secured data through the internet using heterogeneous communication stack implementations. Since there is no "one-size-fits-all" solution, devices customization is commonly adopted. Also, the selection of the protocol stack will directly affect the requirements of the device's hardware. This takes to a large number of different technologies and protocol standards in use and consequently to a device connectivity problem, where several wireless candidates compete for the leading technology to be used [5].

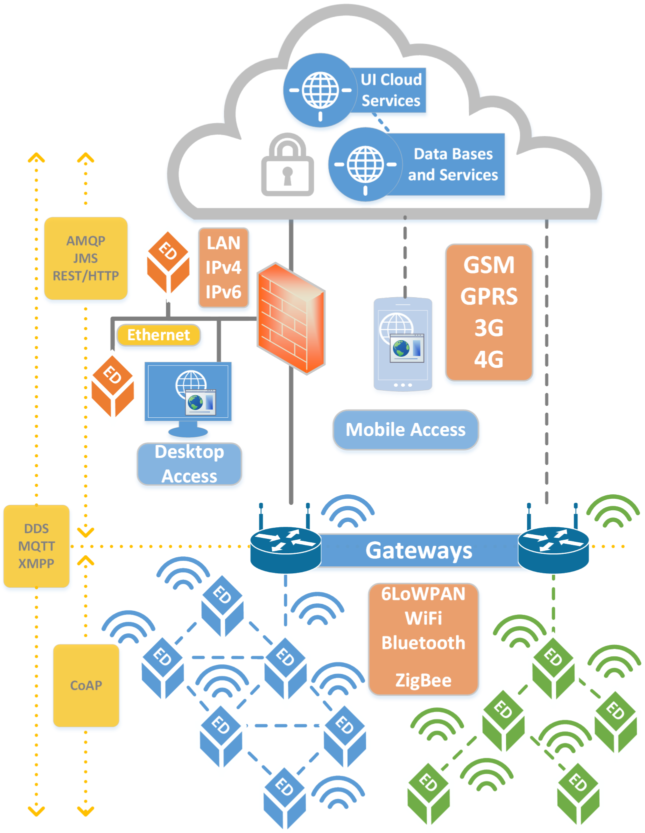

Figure 1. IoT Connectivity Space problem

Concerning the adoption of wireless technologies for IoT, three emerge as the main contenders to maximize deployment, such as IEEE 802.11/Wi-Fi, IEEE 802.15.4/ZigBee and Bluetooth. Heterogeneous implementations with different standards and technologies will have to coexist. The way the IoT devices implement their selected technology will affect the application area and thus, the device's interoperability and connectivity. Fig. 1 (adapted from [6]) illustrates the IoT connectivity space problem (proprietary technologies are not displayed as they are considered "non-standard proprietary standards" and they tend to disappear as new open industry accepted international standards). Not only the wireless nature of the IoT devices is affect, but the whole protocol stack. From the sensor to the cloud, this multitude of standards in use will cause more variability in designing IoT systems [7]. The way the industry and academia will address this problem goes through the implementation of open communication standards and the use

978-1-4673-7929-8/15/$31.00 © 2015 IEEE

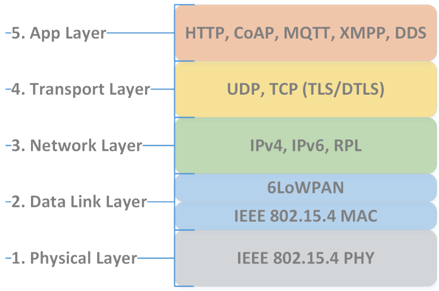

Figure 2. IoT-ARM Communication Stack

of semantic interoperability [8]. Despite of being out of our scope by now, this last approach will be addressed in the future.

This work in progress paper goes beyond state-of-the art presenting an FPGA-based solution for the IoT. The presented work will focus on the connectivity and the interoperability of the edge-devices, by implementing an IoT protocol stack on cost-effective SoC FPGA platform with an integrated hardcore processor. This solution aims to increase the performance and thus improve the efficiency of the edge-devices by implementing design choices, mainly by offloading critical software features (OS or protocol stack) to hardware. This approach aims to evaluate, as a proof-of-concept, future efficient SoC implementations with the new improvements and added features. As technology moves forward, new low-power FPGA platforms may allow to turn the proposed system into a low power and customizable FPGA-IoT platform.

## II. IOT-Architectural Reference Model Overview

With so many big players fighting for their market share, "standard wars" can't be avoided, as each distinct implementation demands specific and suitable protocols to be used. Nevertheless, using a proper reference model which specifies a set of recommended technologies to be used, will lead us to a general consensus where all the existing standards can coexist. In this context, IoT-ARM [9], which does not prescribe any static architectural model, provides guidance for the development of architectures for new IoT systems, aiming to improve interoperability between all the existing IoT devices regardless the vendor or the hardware configuration. This framework is set by several sub-models: Domain, Information, Trust, Security, Privacy and Communication Model. The main focus of our work goes into the Communication Model, since the connectivity and the interoperability play a key role on the IoT devices.

### *A. Communication Model: Connectivity and Interoperability*

All IoT devices must be able to connect seamlessly, but there is no wireless technology that can efficiently serve all requirements along an entire network. The interoperability can be achieved through standard layered protocols [10], where well known defined standards are used.

Fig. 2 illustrates the proposed communication stack for the IoT devices. The layered stack, following a 5 layer OSI model, provides interoperability at different levels, according to the devices' capabilities. The IEEE 802.15.4 standard provides the PHY and MAC layer. This standard, designed for low-rate wireless personal area networks (LR-WPANs) has been widely adopted by well-known technologies like Bluetooth and ZigBee, and proved to be the best for the physical medium as it provides low data-rate radio links on the ISM bands with low power capabilities.

The internet lives on the network layer, connecting any single device with a single unique IP address. With the overloaded IPv4 it is impossible to address the billions of the IoT devices as it only provides 32 bits for unique IP addresses. The new IPv6 is the key for IoT as it provides 128 addressing bits, offering efficient peer-to-peer communication and highly scalable networks [11]. The IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) was specially developed to be used with the LR-WPANs devices and adapted to be used over the IEEE 802.15.4 frames. The 6LoWPAN standard provides a compressed header usable for mesh routing, enabling interoperability at the 3rd layer of the proposed model by implementing the RPL routing protocol (IPv6 Routing Protocol for Low-Power and Lossy Networks). This latter was proposed by CISCO and released by the IETF, allowing the network to be able to configure simple or complex network topologies, like tree, star or mesh.

On the 4th layer (transport layer) UDP or TCP protocols can be used, however, the message control mechanism used by TCP can be hard to handle by the resource constrained devices, turning the UDP the most suitable protocol for IoT. Because it is a connectionless protocol, it has no handshaking dialogues and the message exchange delivery will be controlled from the application side, when needed. For the 5th layer (application) new web protocols have been proposed, as the traditional technologies like HTTP, can't be efficiently used by the IoT devices. New protocols such as CoAP, XMPP and MQTT, are getting position in the IoT space. They provide unique features for different and specific needs, according to end applications: e.g., CoAP uses UDP while XMPP and MQTT runs over TCP [5,6].

## III. IOT-based FPGA Architecture

Aiming to improve the performance of the IoT communication stack, a FPGA-Based IoT architecture is proposed. This architecture, to be used by end-devices and further by the gateways, will take the benefits of the new System-on-Chip (SoC) programmable devices providing on the same chip a hardcore processor and FPGA fabrics.

In a general way, according to the final application requirements, the system designer must select different design tactics in order to meet the application requirements. For this work, tactics aiming to increase systems performance will be explored, by selecting best design choices, e.g., offloading to hardware in this case. For the proposed architecture the hardcore processor will run the Contiki OS, the open-source operating system for IoT, and identified critical communication features will be deployed on FPGA fabrics in order to increase

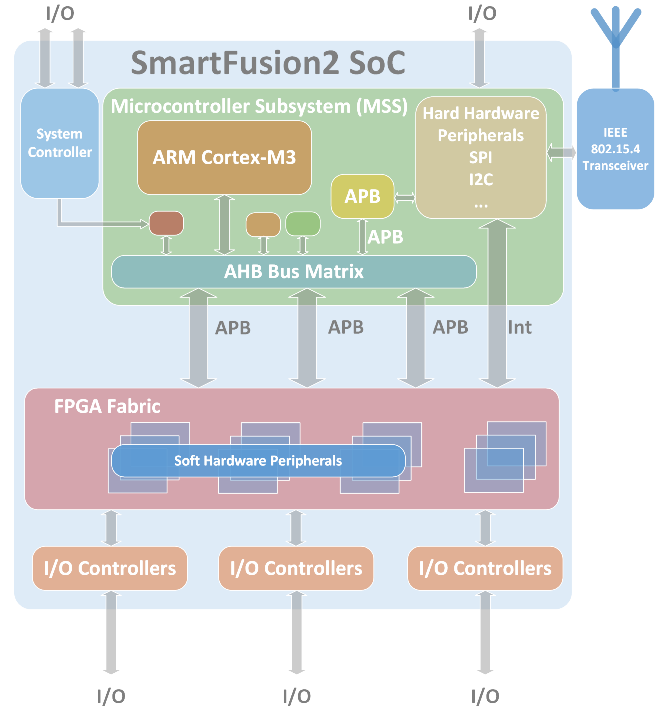

Figure 3. System block diagram

the overall systems performance. Fig. 3 shows the proposed system architecture. It is based on a Smartfusion2 SoC from Microsemi which is mainly composed by an ARM Cortex-M3 processor and the FPGA fabric for specific hardware implementations. An IEEE 802.15.4 RF transceiver is connected to the UART bus to enable the device's communication and allows as well message interception for further processing.

Customized embedded systems with custom hardware profiles are becoming very popular as they provide efficient and specific hardware blocks with specific tasks aiming to preserve the resources of the host CPU for other important tasks, e.g., by offloading the packet processing functions. On important network scenarios, packet processors became very useful as they can perform peculiar tasks in order to provide a fast response to the increasing network traffic loads. They can act on data inspection, extraction, processing, manipulation before the received data reaches the application or the main processor. These FPGA accelerators can implement any layer of the communication stack, usually performing critical and well known mature functions, i.e., functions that are infrequently updated. These dedicated hardware blocks usually perform faster and does not lose flexibility of the applications. A good example of dedicated hardware for network purposes is presented in [12], where a TCP/IP hardware implementation is described, performing seven times better in terms of energy when compared with software implementation.

For the communication stack the Contiki-OS was selected. It provides a full communication stack as suggested in the Fig.2 with other recent low-power wireless standards specially designed for the IoT devices. The software is available as open-source, making it a good choice for the presented work.

## IV. Preliminary Results

The evaluation process was conducted on an in-house evaluation board featuring the SoC CC2538 from Texas Instruments, running at 32MHz, since the porting of Contiki OS for the SmartFusion2 SoC is on-going. As the evaluation is at feature-level and both SoCs have the same microcontroler architecture (ARM Cortex-M3), similar results are expected, with just small differences due to different micro-architectural aspects.

To evaluate which features are suited to hardware offloading, specific microbenchmarks were performed. The selected microbenchmarks encompass some features of the implemented IoT stack, from the MAC layer, till the Transport layer, which include:

- 1 - Delay from read an unprocessed IEEE 802.15.4 frame from the radio RX buffer (FIFO) until it is delivered to the upper layer. The selected API was the MAC broadcast receive function: ***broadcast_recv()***
- 2 - Delay from write a MAC layer broadcast packet, frame it in an IEEE 802.15.4 frame and transmit it to the Radio TX buffer: ***broadcast_send()***
- 3 - Execution time for clearing the IEEE 802.15.4 RF Channel before a packed is sent: ***channel_clear()***
- 4 - Delay from read a processed IEEE 802.15.4 frame from the Buffer until it is read by the UDP transport Layer. This feature assumes an existing packet on the buffer, and for the sake of simplicity, the delay caused by feature 1 is not considered. ***uip_newdata()***
- 5 - Delay from create an UDP packet, frame it in an IEEE 802.15.4 frame and transmit it to the Radio TX buffer. Delay caused by feature 2 is also not considered. ***uip_udp_packet_send()***

Table I
Features Performance Evaluation Results

| *Features* | ***Contiki-OS 2.7*** | |
|---|---|---|
| | $\mu$ | $\delta$ |
| ***broadcast_recv()*** | 103232 | ***309*** |
| ***broadcast_send()*** | 234432 | ***677*** |
| ***channel_clear()*** | 22878 | ***1796*** |
| ***uip_newdata()*** | 232916 | ***1880*** |
| ***uip_udp_packet_send()*** | 651632 | ***30257*** |

Table I presents the obtained results from all the performed microbenchmarks. The presented results give the mean value ($\mu$) of each task (in terms of CPU clock cycles) and the standard deviation ($\delta$). For the ***broadcast_recv()*** and ***uip_newdata()*** functions, the given result suggests that for a proper frame filter implementation in FPGA fabric, the measured delay can be totally removed from the CPU execution time, as the CPU is not interrupted by the radio RX buffer and the packet is processed and dropped prior reaching the CPU. For the ***broadcast_send()*** and ***uip_udp_packet_send()***, the results suggest that the given execution time can be reduced if these features were implemented in FPGA fabric. With dedicated hardware the selected tasks can perform faster and the CPU can be switched to energy saving modes earlier. The ***channel_clear()*** has a big standard deviation and, as it is highly depending on the RF state, it is hard to predict its execution time. Therefore, this feature will be left untouched and performed in software.

## V. Research Roadmap

Further work will focus on the development of network packet processors for IoT devices. The main goal is to accelerate the processing of the incoming IEEE 802.15.4 frames and its content in order to increase system's performance. Also, by using an external RF transceiver coupled with the system's communication bus, packets can be pre-processed in parallel with CPU execution. Delegating specific tasks to dedicated hardware, the CPU can spend more time in sleep mode, being only interrupted when the received packets need to be forwarded to the higher layers for further processing.

For the IEEE 802.15.4 MAC frames, we propose a frame filtering scheme, where IEEE 802.15.4 important fields such as the PAN_ID, SHORT_ADDR, EXT_ADDR, can be processed before the packet reaches the CPU, avoiding unnecessary packet delivery when the packets do not match the PAN ID or the destination address. Although this feature may be present on some newest RF transceivers, it is limited to a small fixed number of preconfigured addresses. Because the memory resources of the proposed platform is considerably larger than the RF transceiver, we can increase the available amount of memory for addresses, and also extend the current filtering features. If the content of the incoming IEEE 802.15.4 messages is processed, other filters and features can be implemented. As the suggested IoT stack is fully implemented by the proposed system, incoming packets can be processed at other layers such as: (i) the Data Link Layer (6LowPAN), where the IPv6 headers (containing the addressing information and IPv6/UDP headers) can be pre-filtered prior delivery to the CPU; (ii) the network layer, where an RPL routing improvement can be also suggested, with some routing features like the routing tables and the RPL advertisement scheme can be offloaded to the FPGA fabric in order to assist the network packet processors.

From a different perspective, research will continue towards the development of a secure gateway device. Since security is emerging as a new dimension in system's design, and gateways are critical infrastructures that establishes the bridge between the PAN and the internet, the ideas and the concepts behind our proposed edge solution will be scaled and simultaneously extended with security designed from the outset. ARM Trustzone Technology [13] will be exploited to manage critical software stack components among different levels of security. For instance, Contiki OS will be refactored to run the small kernel and the service manager in the secure world side, and services (e.g., communication protocol - TCP/UDP stack) and applications on the normal world side. Xilinx ZC702 Evaluation Kit will be used as target platform, since it provides a Zynq-7000 SoC with a Processing System formed around a dual-core ARM Cortex-A9 (supporting TrustZone) and Programmable Logic (i.e., FPGA).

## VI. Conclusion

The Internet of Things, long ago ceased to be purely a future vision and is now a reality. The billions of smart devices, using different mechanisms to communicate among them and with the Internet, lead to different standards and technologies to coexist. To deal with this level of heterogeneity while providing connectivity and interoperability among the existing wireless devices, the IoT-ARM specifies a full communication stack.

The presented work proposes the prototyping of an FPGA-based edge device for IoT, focusing on the connectivity space problem. It is described the planned implementation for an IoT protocol stack under a cost-effective SoC FPGA, and presented which features appear to be good candidates (based on the preliminary results) to offload to hardware.

## VII. Acknowledgements

Tiago Gomes is supported by FCT - Fundação para a Ciência e Tecnologia (grant SFRH/BD/90162/2012). This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope: PEst-UID/CEC/00319/2013.

## References

[1] M. Bal, "Industrial applications of collaborative wireless sensor networks: A survey," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, June 2014, pp. 1463–1468.

[2] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, Oct 2009.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *Internet of Things Journal, IEEE*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[4] ARM. (2014) From Sensor to Server. [Online]. Available: http://www.arm.com/markets/internet-of-things-iot.php

[5] Micrium. (2015) IoT for Embedded Systems: The New Industrial Revolution. [Online]. Available: http://micrium.com/iot/overview/

[6] A. Foster, "White paper: Messaging Technologies for the Industrial Internet and the Internet of Things," PrismTech, Tech. Rep., 01 2015.

[7] C. Cees Links, "White paper: Wireless Communication Standards for the Internet of Things," GreenPeak Technologies, Tech. Rep., 01 2015.

[8] J. Kiljander, A. D'Elia, F. Morandi, P. Hyttinen, J. Takalo-Mattila, A. Ylisaukko-Oja, J.-P. Soininen, and T. Cinotti, "Semantic interoperability architecture for pervasive computing and internet of things," *Access, IEEE*, vol. 2, pp. 856–873, 2014.

[9] M. B. et all, "Deliverable D1.5 - Final architectural reference model for the IoT v3.0," Internet of Things - Architecture, Tech. Rep., 01 2015.

[10] S. L. Inc., "Overcoming Challenges of Connecting Intelligent Nodes to the Internet of Things," Silicon Laboratories Inc., Tech. Rep., 01 2012.

[11] I. Cisco Systems, "White paper: Integrating an Industrial Wireless Sensor Network with Your Plant's Switched Ethernet and IP Network," Cisco Systems, Inc, Tech. Rep., 01 2009.

[12] N. Maruyama, T. Ishihara, and H. Yasuura, "An rtos in hardware for energy efficient software-based tcp/ip processing," in *Application Specific Processors (SASP), 2010 IEEE 8th Symposium on*, June 2010, pp. 58–63.

[13] S. Pinto, D. Oliveira, J. Pereira, N. Cardoso, M. Ekpanyapong, J. Cabral, and A. Tavares, "Towards a lightweight embedded virtualization architecture exploiting arm trustzone," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, Sept 2014, pp. 1–4.