Future Internet and Reconfigurable Computing: Considerations on Flexibility and Security

D. G. Mesquita and P. F. Rosa

***Abstract*—The Future Internet is expected to support services in both existing and new scenarios, in terms of mobility, quality, scalability and security, among other. In this work we present how Reconfigurable Computing (RC) may contribute to build Future Internet (FI) flexibility and security. Therefore, we discuss some aspects of FI initiatives that can be addressed by Reconfigurable Computing. Then we show some features of the Reconfigurable Computing enabling technology – FPGA – which can help to build a more flexible and secure Future Internet. The concluding remarks concern the need to bring together FI and RC researchers.**

***Keywords*—Future Internet Architecture, Reconfigurable Computing, Security.**

I. Introduction

THE Internet growing success is due, in part, to its achitectural simplicity. Designed in the 70's and consolidated in 1981 with the request for comments 791 which has standardized IPv4 - the Internet Protocol version 4 [1], the network of networks has kept, through time, the flexibility needed to incorporate new technologies in order to support a whole myriad of applications.

IPv4 initial specifications established its independence from the underlying layers, and also concerning the host architecture, as well as universal connectivity through entire network, point-to-point acknowledgements and standardized application protocols. Another success motive which should be highlighted is the Berkeley University decision of adopting the TCP/IP implementation in its 4.2 BSD Unix in 1983, and make its source code available as public domain software.

However, the TCP/IP idealizers could not foresee the reach which this network would achieve in the next decades. As can be seen in Fig. 1, in 1970 there were only 9 hosts on the Internet, while the current number is significantly greater.

If, initially, the concern was to interconnect a few computers in research centers, today, 35 years after the RFC:791, the Internet must cope with mobile devices and with the Internet of Things, where devices can "talk" to each other without direct human intervention. Considering the numbers related to this phenomenon, in 1985 there were 1,000 hosts connected, in 1995 it was 3,000,000 [3] and, today, there is approximately one billion hosts linked to the Internet [4]! Fig. 2 helps to visualize the Internet growth, and gives an idea of the complexity concerning the hosts interconnection of such numerous devices, especially if compared to Fig. 1.

D. G. Mesquita, Universidade Federal do Pampa (Unipampa), Santana do Livramento, RS, Brasil, mesquita@unipampa.edu.br

P. F. Rosa, Universidade Federal de Uberlândia (UFU), Uberlândia, MG, Brasil, frosi@ufu.br

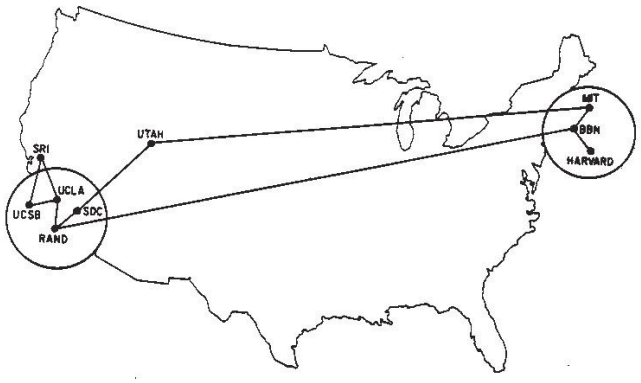

Figure 1. The Internet in 1970. Source: [2]

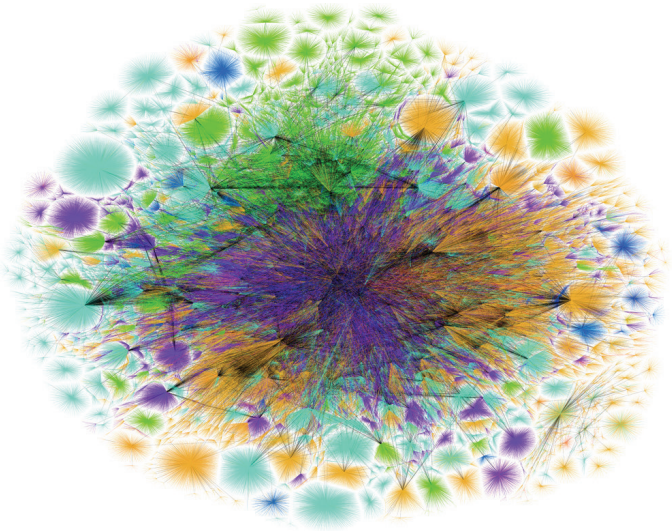

Figure 2. The Internet in 2015. Source: [5]

These numbers leads to the first IPv4 limitation: the scarcity of network addresses. Because of this limitation, many companies implement the network address translation (NAT) in order to map different private IPv4 addresses into an unique public IPv4 address. This technique has helped in conserving public IPv4 addresses, but also have some drawbacks. The NAT does not support the security standards of the network layer, and does not support mapping all protocols of superiors layers. Besides, the address configuration in various devices, whether static or dynamic, should be much simpler than the current way.

Another current Internet problem relates to quality of service (QoS). Despite IPv4 supports QoS, it has only 8 bit of the Type of Service (ToS) field and payload identification to performe it. The IPv4 ToS field has limited functionality and the packet identification is not possible when the datagram packet is encrypted [6].

Talking about cryptography, when the RFC:791 was published, almost none of the current security threats were anticipated. An IPv4 extension, called IPSec [7] was suggested in order to protect data transmitted through Internet, avoiding data visualization or modification by unauthorized people. However, the IPSec is not an Internet built-in protocol and, in many times, its implementations are proprietary.

Meanwhile, with the growing threat of cyber attacks, the security as a whole, and the cryptography - as its support pillar, became a central matter about the future of the Internet.

The scientific and industrial communities are aware of these problems and are proposing changes in the Internet for some time. Several initiatives have been proposed aiming to develop Future Internet Architectures. Some examples are the projects FIBRE (Brasil), FIRE (Europe), NETS-FIA (USA) and AKARI (Japan), among others. More information about this subject are found in Section II.

These initiatives have a lot in common, but it is worth to highlight two important aspects: flexibility and security. Flexibility is necessary both to conduct the transition from the current to future Internet as for the Internet's purpose itself, much more focused on associated services and content type rather than mere packet transmission. Security is another aspect that encourages the development of a new Internet. Cyber attacks are becoming more and more sophisticated and it threatens the economic order. If the current Internet was not conceived to deal with such threats, the next generation of Internet need to have security as a major concern.

Given this context, one technology - now mature - that could be largely used in the very conception of the Future Internet in order to build network flexibility and security has been neglected. This technology is the Reconfigurable Computing, which is based on field programmable gate arrays (FPGA). FPGA have features which can meet both the requirements of flexibility as the security for Internet next generation.

The idea of a reconfigurable device was first conceived by Gerald Estrin in 1960 [8], but only in 1985 they became commercially available [9]. Fig. 3 shows the experiment made as proof of concept by G. Estrin [10]. Besides all technology limitations at that time, the idea of a processing element having a fixed part and another with a flexible hardware, adaptable for any given application, was genius. The hardware flexibility was achieved through the inclusion or removal of specific hardware modules and by reconfiguring its interconnections. If in 1960 the task was manual and fastidious, today, with submicron fabrication technology with a tremendous evolution of the Electronic Design Automation (EDA) tools, FPGA have became an unavoidable solution, when searching for a good compromise among performance, low cost, low power, flexibility and security

Today there is FPGA based systems in sectors such as aerospace, automotive, defense, industrial automation and data-

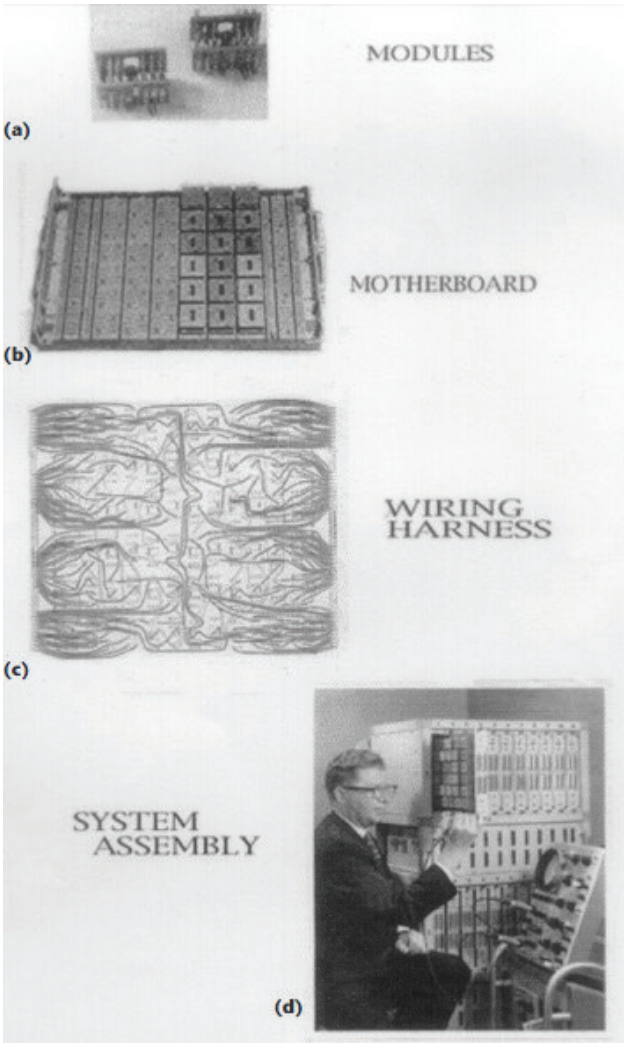

Figure 3. Basic reconfigurable modules (a), motherboard (b), wiring harness for the motherboard (c), and system assembly constructed for the supervisory control and transfer path between the fixed and variable structure computers. The author is shown using an oscilloscope probe to observe electrical activity of the system assembly (d). Source: [10]

centers, among many others. There are even FPGA prototyping boards designed specifically to support network research and development [11], [12].

This article aims to approach these two investigation fields Future Internet and Reconfigurable Computing, discussing how this approximation could lead to a next generation of networks more flexible and secure.

In order to do this, the article brings in its second Section an overview of Future Internet initiatives, with some project examples and a brief discussion about how flexibility and security are important to these enterprises. On the third Section, this article argue about the enabling technology of Reconfigurable Computing: the FPGA. The third Section shows briefly how FPGA works and highlights how its architecture helps to achieve flexibility and security. Finally, the last Section draws a roadmap in the direction of a convergence between researchers in "Future Internet"and Reconfigurable Computing, so the next generation of network can be more flexible and secure.

II. Future Internet Architecture

As mentioned before, there are many Future Internet Architecture initiatives around the world. In this Section I present an overview of a few of those initiatives.

A. Future Internet Research and Experimentation

FIRE (Europe): Future Internet Research and Experimentation is an initiative launched and funded by the European Commission that has been growing since its inception in 2010 with the ambition of being Europe's Open Lab for Future Internet research, development and innovation. One (of many) interesting aspects of the FIRE is the importance given to experimentation as development methodology. The white paper [13], a document on the FIRE initiative, reports some experiments that pushes the knowledge frontiers concerning connectivity. Among the examples there are the investigation of Facebook on economics of privacy; the Netflix experimentation platform to ensure optimal video streaming delivery with minimal playback interruption; Smart Cities using diverse network applications in fields such as transport, energy and environmental management.

Those well succeeded development strategies based on experimentation led the FIRE organizers to adopt a research methodology named experimentation-driven. Another interesting characteristic of the FIRE initiative is the engagement of the industry partners both in financing as in developing projects. Still in [13], the security theme is mentioned many times, from standards for mobile communication untill security and privacy in smart spaces, passing by aspects of trustworthiness, dependability and border control in autonomous cooperative robots.

B. Future Internet Brazilian Environment for Experimentation

Helped by the European Commission, Brazil has developed an experimentation environment - a testbed - which works as a large scale laboratory. The goal of this testbed is to serve as research and development infrastructure so students, researchers and industry may test new network applications and architectures. The project is named Future Internet Brazilian Environment for Experimentation (FIBRE)[1], and it is composed by eleven experimentation islands, sheltered in universities and research institutes. Each experimentation island has a set of equipments which supports experiments in both wired networks as wireless. Those islands are connected by an network on the brazilian RNP backbone, comprised of two network separate layers: a control plane and an experiment plane [14]. The FIBRE community is active, with recent publications, as [15] and [16]. Some projects related to FIBRE include OpenFlow [2] and Software Defined Networks (SDN). In the context of FIBRE, these concepts are related to FPGA implementations, as can be seen in Fig. 4.

[1]http://www.fibre.org.br/

[2]https://www.opennetworking.org/sdn-resources/openflow

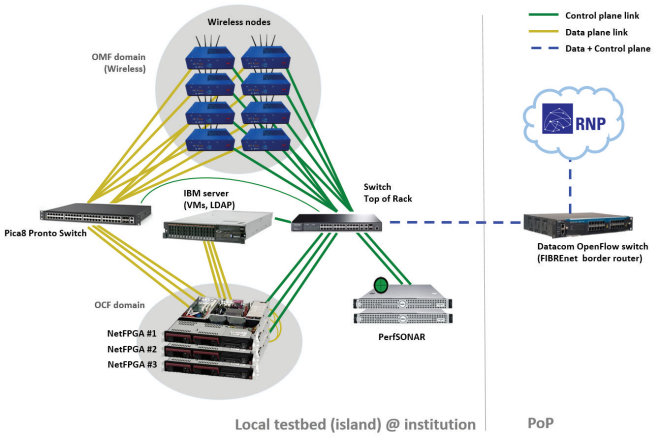

Figure 4. A hardware view of a typical FIBRE island. Note the NetFPGA as a part of the infrastructure. Source: [17]

C. NETS-Future Internet Architecture

NETS-FIA (USA): Recognizing the need for a secure and highly dependable information technology infrastructure and building on NSF's on-going investments in network science and engineering, the Directorate for Computer and Information Science and Engineering (CISE) has formulated this program to stimulate innovative and creative research to explore, design, and evaluate trustworthy future Internet architectures. The NETS-FIA objective is to engage the research community in collaborative, long-range, transformative thinking - unfettered by the constraints of today's networks yet inspired by lessons learned and promising new research ideas - to design and experiment with new network architectures and networking concepts that take into consideration the larger social, economic and legal issues that arise from the interplay between the Internet and society [18]. One of the many projects derived from the NETS-FIA initiative is the Named Data Networking (NDN) [19]. The NDN aims to develop a new Internet architecture keeping the strengths of the current one, and addressing its drawbacks. Its main aspect is to name the contents instead its location. For instance, the current Internet secure the data container, while the NDN secures the content. This is an architectural choice which decouples data confidence from hosts confidence. The project studies the technical challenges to be overcomed in order to validate the NDN as Future Internet: routing scalability, network security, content protection and privacy. Thus, as the previous initiatives, the NETS-FIA is strongly based on experimentation and on the prototyping of protocols. Also, there is an emphasis on the security theme.

D. AKARI

AKARI: The japanese AKARI[3] Architecture Design Project aims to implement the basic technology of a new generation network, developing a network architecture and creating a network design based on that architecture. Its philosophy is to pursue an ideal solution by researching new network

[3]The AKARI project aims to be "A small light (akari in Japanese) in the dark pointing to the future."

architectures from a clean slate without being prevented by existing constraints. The project principles are "crystal synthesis", "reality connected"and "sustainable and evolutionary". The explanation follows:

- *Crystal synthesis* means that the project must remain simple, even when integrating different functionalities.
- *Reality connected* stand for to separate physical and logical infrastructures.
- *Sustainable and evolutionary* represents the properties of self-organization and self-distribution, being flexible and open to future changes.

Projects related to AKARI deal with different aspects networks, and among them stand out those concerning data-centric networking systems [20].

Although the AKARI project was discontinued in 2013, the National Institute of Information and Communications Technology (NICT), from Japan, continued to stimulate research on the post-Internet network, whose goals include addressing current network issues such as reliability and security [21]. According to the authors, this new network must also be flexible and sustainable.

E. Section Summary

There are two major concepts standing out from the subsections above: flexibility and security.

Flexibility is needed to perform experimentations, but also to support future changes on the Internet architecture, so it can evolve. Nowadays, the flexibility is deeply related to SDN and to the data-centric networking systems. Security is essential to the economic success of any network [22]. As in the current Internet almost all security is performed at the application level, the Future Internet must ensure data content at lower levels. Fig. 5 depicts the concern with these aspects.

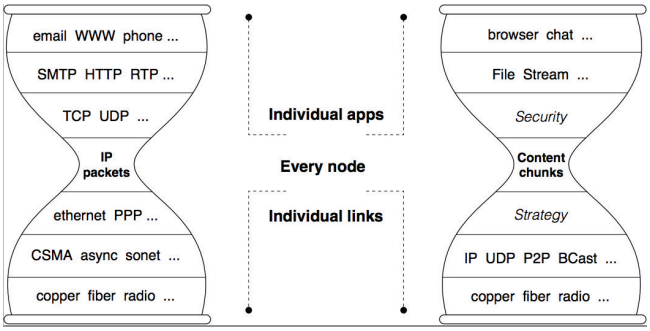

Figure 5. The Internet and the NDN hourglass Architecture. Source: [23]

Fig. 5 compares the current Internet (at the left side) with the next generation of Internet proposed by the NDN team (at the right site). The focus is no longer at the packets communication, but changes to content distribution. Rather than “fixed” protocols, the Future Internet deal with strategies in order to guarantee the data arrival at its due destination. Concerning security, a neglected feature at the original Internet architecture, receives a spotlight at the Future Internet Architectures as proposed by the Named Data Networking team.

The next Section presents the FPGA architecture characteristics that may contribute to achieve flexibility and security (and performance) in projects concerning Future Internet Architectures.

III. Reconfigurable Computing

Reconfiguable Computing is a relatively new computational paradigm which fills the gap between the software flexibility and the specific hardware performance [24]. As mentioned at the Introducion, the reconfigurable computing enabling technology is the FPGA. But unlike has been shown in Fig. 3, the current FPGA are fabricated in nanometric scale and the process of design and prototyping is all assisted by sophisticated Electronic Design Automation (EDA) tools, making possible a short time-to-market and a longer product life-cycle. Fig. 6 depicts the position of systems based on FPGA regarding those based on CPU (microprocessors) and those based on Application Specific Integrated Circuits (ASICs). As can be seen in this abstraction, FPGAs fill a gap between performance and flexibility. Yet, if we change the Y axis from flexibility to energy efficiency, the positions remain the same.

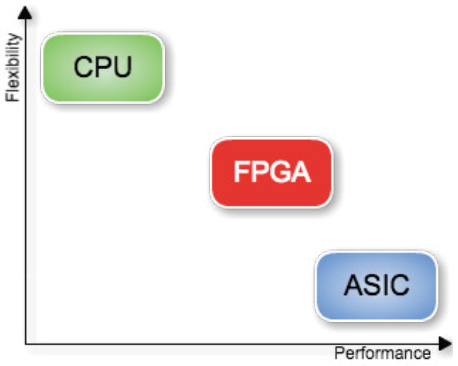

Figure 6. FPGA bridging the gap between microprocessors and application specific processors. Source: Adapted from [25]

Another important factor favouring the FPGA is its cost. Although for parallels applications its performance approaches to the ASIC, as the FPGA is an off-the-shelf product, there is not the Non-Recurring-Engineering (NRE) costs, tipically high for ASIC projects. Moreover, as the design time for FPGA is only a fraction of those related to ASIC, the development costs of systems based on FPGA are significantly lower than those associated with ASIC. Meanwhile the FPGA time-to-market is shorter [26], fact that can improve the profitability of those who adopt this technology. However, perhaps the main interest concerning FPGA is its reconfigurability, since this feature allows the hardware to be modified at the logic function level. This hardware flexibility may lead to a longer product life-cycle, since the hardware functionality can be upgradable without change any physical element.

In order to support these assertions, the next subsection gives an overview of the FPGA architecture. After, in the following two subsections two themes are discussed: (i) topics about FPGA used in rapid prototyping of digital systems, specially concerning networking and (ii) FPGA used to improve the security infrastructure, in particular for cryptographic applications. The Section ends by establishing the link between

the topics mentioned right above and those covered at the end of Section II: network flexibility and security.

A. FPGA

This subsection is a brief overview about FPGA. If you are interested in a detailed description of its functionality, please refer to [27]. For an extended survey on Reconfigurable Computing, We recommend to read [28]. For an up-to-date summary of the knowledge-base on FPGA architecture, tools and systems I suggest the article [26]. Finally, there are Reconfigurable Systems that are not based on FPGA. These are not covered by this work, but are discussed on [25].

FPGA are digital integrated circuits whose functionality is not pre-defined and can be changed after its fabrication. Basically it is an array of programmable logic elements, interconnected by a mesh of also programmable routing resources, which functionality can be field programmable. Fig. 7) depicts the FPGA basic architecture in a high level of abstraction. Depending on the fabrication technology - anti-fuse, Electrically-Erasable Programmable Read-Only Memory (EEPROM), static RAM (SRAM), magnetic RAM (MRAM) - the FPGA can be programmable once or many times. The most common is the SRAM based FPGA, which can be reconfigured numerous times [27]. The magic behind the FPGA is the capability of receiving any functionality that can be algorithmically described. This is possible because the logic elements composed by look-up tables (LUTs), multiplexers, flip-flops and some logic gates. Each LUT can be used as a small RAM, or, most commonly, as a logic function. Typically a FPGA has thousands of logic elements, communicating through interconnection resources composed by wires and muxes. Isolated, a logic element cannot do much, but with the rich interconnection, complex logic functions can be performed. The designer may describe the FPGA functionality using a hardware description language (tipically VHDL or Verilog). The codification style can be algorithimically (behavioural) or structural. The most common is a mix of both, emphasizing the structural one (for the synthesis sake). Once the description is finished, an EDA tool helps to transform it into a bitstream, which is the configuration file format for the FPGA. This process is called synthesis. There are intermediary steps, but most of it is automated. Fig. 9 (left) helps to visualize this process.

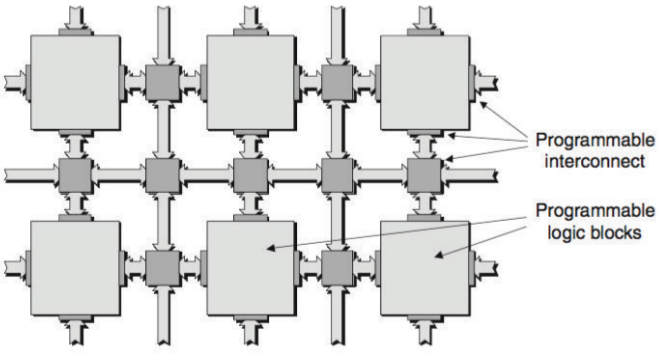

Figure 7. A simplified view of a generic FPGA. Source: [29]

Fig. 7 is a mere abstraction, because, actually, FPGA architecture include I/O pins, fast interconnection network, clock trees, multiply and accumulate (MAC) elements, memory blocs, small DSPs, and even, sometimes, one or more microprocessors (softcores or hardwireds). By adding some general purpose processing as small CPUs and DSPs into the FPGA matrix, the Reconfigurable Computing is improved, as it approaches the best of both worlds. Thereby, it is possible to see the FPGA as a co-processor of a conventional microprocessor (which is, in its turn, into the FPGA). It is also possible to imagine the FPGA as processing unity attached to the microprocessor via shared memory. On a third scenario, the FPGA can act as a stand-alone processor. Fig. 8 depicts those three possibilities of Reconfigurable Computing.

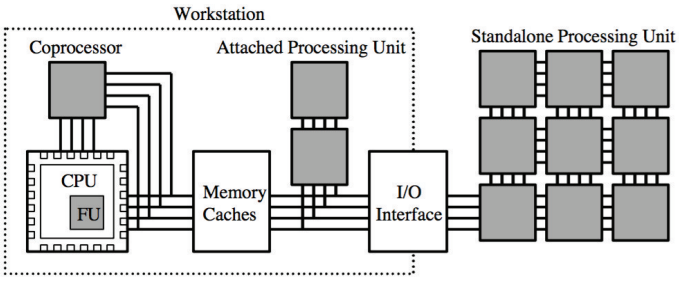

Figure 8. Different coupling scenari between Reconfigurable Computing (RC is shaded) and conventional computing (CPU). Source: [30]

However, each project must carefully analyse what is more suitable to its goals, because each scenario has strengths and weakness.

On the one hand the tighter the integration between the reconfigurable hardware and the microprocessor, more often the reconfigurable fabric can be used by a given application, due to the low communication overhead. On the other hand, the hardware is unable to operate for large time slices without the CPU intervention. Often in this case, the amount of logic elements available to the application is reduced, as part of it is used to implement the CPU itself. The more loosely coupled with the CPU, more room to explore the application parallelism, but with a penalty of increasing the communication overhead.

Regarding design costs of a digital system, the design flow for FPGA is simpler than that for ASIC, as can be seen in Fig. 9. Although both begin similarly, with the functional specification phase, the hardware description (with VHDL or Verilog) and a behavioural simulation, the ASIC flow needs static timing analysis and equivalence checks with the foundry parameters. Also, the ASIC flow must include verification of internal deep sub-micron effects (verification on second and third order effects). Concerning the FPGA these verification are previously made by the manufacturer so the end user (in this case the digital designer) do not to bother with it.

Not to mention that, once the flow is exectute for FPGA and the bitstream is generated, just download it into the FPGA on a prototyping board to test it. If something goes wrong, just go back to the HDL phase and start again.

In the case of the ASIC flow, it is not so easy. Once finished the system verification using EDA tools, the physical layout

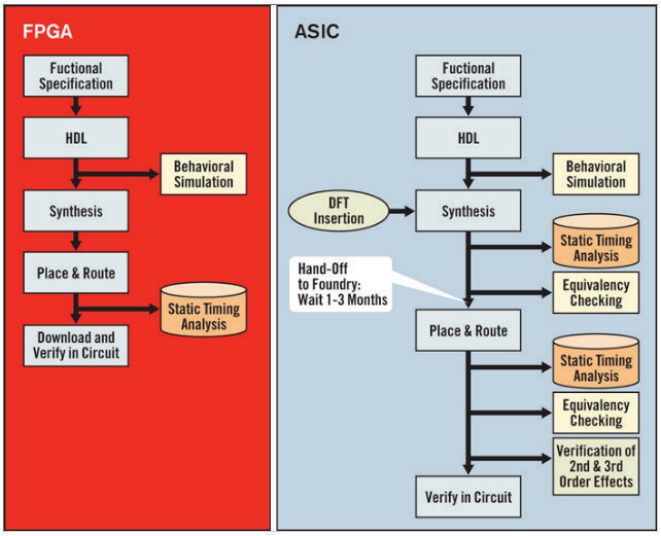

Figure 9. Comparison between FPGA and ASIC design flow. Source: [31]

must be done, and after that there is the post-layout technology checks. If everything is ok, the layout can be sent to the manufacturer (frequently overseas). Once the chip done, it is sent to the encapsulator, and then, sent back to the original designer in order to test. If the test fail, all process must restart. As you can see, it is an expensive process.

To illustrate an FPGA prototyping board and its capabilities, Fig. 10 shows a low-cost board, where a whole digital system can be implemented and tested. Once the goals achieved, the system can be sized to fit its needs, in a specically designed board with only the needed resources, with and FPGA equivalent to that on the prototyping board.

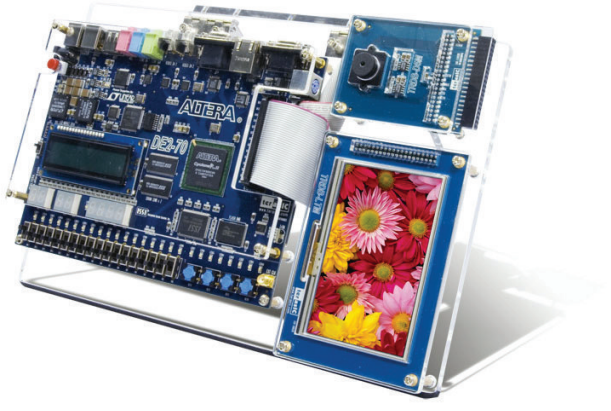

Figure 10. An Altera FPGA Prototyping Platform. Source: [32]

Fig. 10 depicts the I/O richness that can be attached to a FPGA, in the sense of prototyping a variety of applications. For instance, the camera and the display could be used as I/O for a facial recognition on a access control system to a mainframe room, where the heaviest processing algorithm are computed on the FPGA.

Such flexibility, and low implementation cost, besides the high performance when compared to purely software solution, indicate the FPGA as an ideal platform to prototype digital systems. The next subsections gives some examples network and security experimentations using FPGA.

B. FPGA and Networking

A recurring subject in developing the Future Internet is the Software Defined Network (SDN) model. In the SDN architecture, the control plan and the data plan are decoupled, the network inteligence and status are locally stored, and the underlying network infrastructure are abstracted from the application [33]. In short, SDN emphasizes the following characteristics:

1) Decoupling networking hardware and software;
2) Centralized network view and control;
3) Open interfaces between devices on the control plan and the data plan; and
4) Programmability by external applications, i.e., operators, independent sofware vendors and users - not just equipment manufacturers.

Decoupling networking hardware and software allows for centralization of the control portion (named the control plane) while keeping the actual packet forwarding function (the forwarding plane) distributed across many physical network switches. This provides a means to configure, monitor, troubleshoot, and automate a large network built of many discrete hardware components as a single network "fabric."

The centralized control plane can then enable new or different forwarding behaviors and broader, more precise control of traffic flow. Many products that encompass data center fabrics and flow control methods such as OpenFlow leverage this facet of SDN.

The switch required by the SDN model must process packet flows in a performing and secure way. So, basically, there are three elements to consider: performance and security, but also flexibility as preconized by the OpenFlow switches.

The reference [34] is a report on initiatives about SDN switches using different approaches: multicore CPU/GPP; Network Processing Units (NPU) / Network Flow Processors (NFP); PLD/FPGA; Application-Specific standard products; and ASICs. Fig. 11 relates them in a plan composed by the Programability (flexibility) axe and Performance (in Gb/s) axe. We can see a close relation with Fig. 6, where FPGA appears as a half-way between performance and flexibility. This finding is also supported by article [35].

One concrete example of convergence between Reconfigurable Computing researchers and network researchers is the NetFPGA project [36] from the University of Cambridge. This project, based on FPGA technology, provides software, hardware and community as a basic infrastructure to simulation and testing high-speed networks.One key point of this project is to maintain the platform as an open-source project, allowing the reuse of building blocks across various research projects. The current board (NetFPGA SUME) supports up to 100GB/s applications. The NetFPGA SUME shown in Fig. 12 uses a high-density Virtex 7 690T FPGA, supporting high-speed serial interfaces, and its format permits user-expansion.

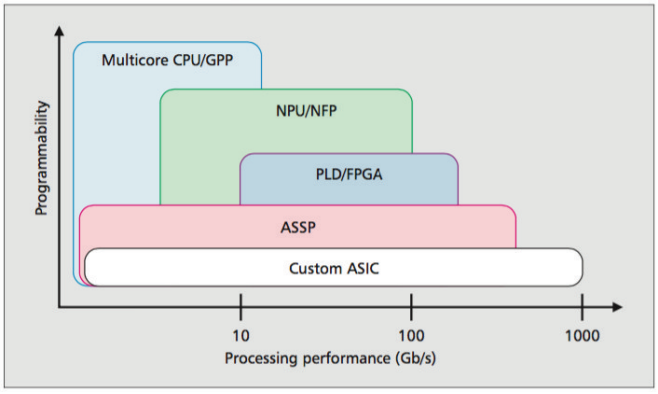

Figure 11. Network processing: performance x programmability Source: [34]

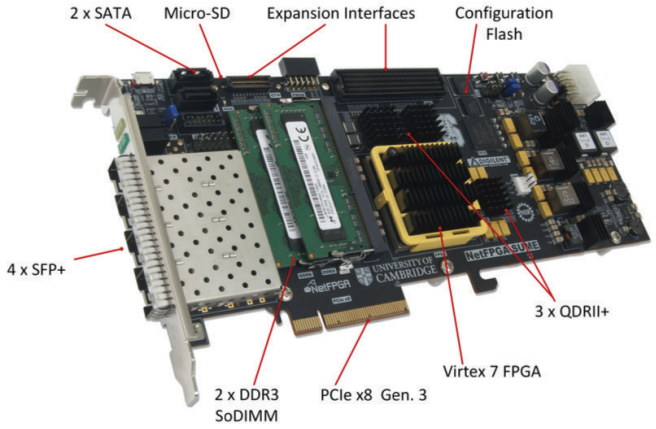

Figure 12. NetFPGA SUME board. Source: [36]

Acording to the project maintainers[4], there currently are more than two hundred scientific publications of results on network research makin use of NetFPGA, across 150 research groups in more than 40 countries. These project includes an open-source network tester [37], a high-resolution hardware based packet capture [38], an evaluation of native load distribuition of ARP-path in data centers [39] and a framework for trust and policy management for a secure internet [40]. In all these works, the common point is the claim that is through experimentation on a real hardware, rather than simulations, is that researchers can reach more assertive conclusions.

Another significant example is the development process of a high availability routing protocol - called HARP [41], researched on the context of the Future Internet project called Entity Title Architecture (ETArch) [42].

At the work presented in [43], the author has found several problems at the VRRP (Virtual Router Redundancy Protocol) that could lead to sensible downtime intervals at the network of a major telecom group. He identified the split brain and the no-brain situations as causes to these downtimes. Then[43] proposes n extension to the VRRP in order to address the problems found. However, when trying to implemente the work in [43], we found some inconsistencies, unobservable without physical experimentation. By using some low-cost FPGA we were able to recreate the virtual router scenario and to observe the failures in VRRP. By constructing a hardware model to validate high-availability protocols, we achieved to develop and test the HARP, solving the no-brain and the split-brain situations. Fig. 13 depicts the solution. The HARP protocol is essentially a finite state machine (at the center of Fig. 13). The surrounding blocks are part of the validation scheme. More information about HARP protocol can be found in [44].

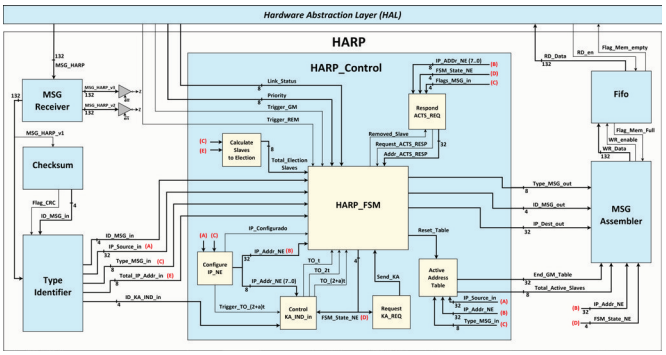

Figure 13. HARP Architecture. Source: Self.

As illustration, the Fig. 14 shows the FPGA boards where the HARP was prototyped. Each board acts as a router, and all three composes a virtual router. Many different scenarios were described and experimented, giving us confidence in our both theoretical and practical findings.

Figure 14. HARP prototype experimentation. Source: Self.

Both situations, the NetFPGA community and numerous research papers, and our own results, demonstrate the potential of FPGA as platform to implement future internet solutions, in particular in cases where the experimentation is part of the research and development methodology, as foreseen in FIBRE, NFS-NET, FIRE and AKIRA initiatives.

C. FPGA and Network Security

As mentioned before, FPGA has a regular and homogeneous internal architecture, with a rich interconnection network. These are important ingredients to compute parallel algorithms, as it is the case of cryptography or pattern recognition. Such algorithm classes are in the basis of network security solutions, as authentication (through public key cryptography),

[4]http://netfpga.org/site/\#/publications/

packets classification, or intrusion detection, only to mention a few examples.

The networks composing the Internet, the current one or the future, need security solutions with the following features [45] :

- Real-Time Protection. For an effective protection mechanism, it is important to achieve line-speed data processing with an affordable cost.
- Flexible Updating. As attackers are creative and their methods are continuously evolving, the protection system must also be adaptable.
- Scalability. This is a critical concern for actual deployment. Many approaches working in small networks have their performance deteriorated in real scale networks. Due to its regular architecture, FPGA can be added in a scalable way.

The features above are well addressed by FPGAs. Its intrinsic parallelism can meet the real-time requirements for packets classification, for instance, at a lower cost than ASICs. Although flexibility also can be achieved with software in CPU or NPU solutions, those does not necessarily meet the performance conditions, meanwhile the ASIC are fast, but expensive and not flexible at all. Once again, FPGA emerge as the alternative which fills the gap among all architectural approaches.

Still regarding these features - performance and flexibility - there is some examples that is worth to take a look:

- *Packet classification* this is one of the fundamental challenges in designing high speed routers. It enables the router to support firewall processing, quality of service differentiation, policy routing and other value added services. When a packet arrives at a router, its header is compared with a set of rules. Each rule can have one or more fields and their associated value, and an action to be taken if matched. To perform the needed comparison to each packed, FPGA has been successfully used [46], [47], [48]. In [49] we find a comparison between packet classification implementations in FPGA, GPP and GPU, with an impressive advantage to FPGA.
- *Pattern matching* is the most important and the most computationally intensive component of a network intrusion detection systems (NDIS). NDIS operates by monitoring network packets and matching them against user-defined rule set. Many successful works has been done using FPGA to perform pattern matching [50], [51], [52]. [53] brings us a comprehensive survey about pattern matching for deep packet inspection, including FPGA implementations.
- *Distributed Denial of Service (DDOS)* and Internet worms attacks are the two major security theats to the network infrastructure [54]. In [55] active scans and filters has been implemented, in order to detect Internet worms and viruses at a multi-Gigabit/second rates, using FPGA. In [56] the authors achieved 400Kilo Packets filtering in a anti-DDOS system based on the NetFPGA 1G. In [57] the authors implemented a real-time detection against DDOS and IDS (Intrusion Detection System) based in FPGA, achieving a throughput of 2 Gigabits per second.
- *Cryptography* is the fundamental component for securing the Internet traffic. However, cryptographic algorithms impose high processing time and efforts that can be a bottleneck to high-speed networks. It has been demonstrated the advantages of using FPGA for cryptographic applications. [58] shows a fast Elliptic Curve Cryptography in FPGA. In [59] the authors present three reconfigurable hardware architectures for modular exponentiation, the main function of the RSA cryptosystem.In [60] we found an efficient FPGA implementation of the AES private-key cryptographic algorithm.

However, FPGA implementations of cryptographic primitives deserve some attention concerning some potential vulnerabilities.

First of all, designers should be advised that any hardware implementation of cryptographic primitives also may be vulnerable to hardware-specific attacks. Even choosing to implement a computationally secure cryptographic algorithm (such as RSA [61] or ECC [62]), cryptographic hardware modules may suffer with direct or side-channel attacks [63], [64]. Direct attacks may attempt to change the implemented logic. To counteract it, FPGA vendors have developed tamper-evidence and tamper-resistance techniques [65],[66], [67]. Concerning SCA attacks, there also some techniques the designer must be aware to use [68].

Fig. 15 summarizes hardware threats and countermeasures. At the middle, the figure shows the abstraction levels, from the lower level, which is the fabrication technology up to the application level. Each level has known threats and corresponding countermeasures.

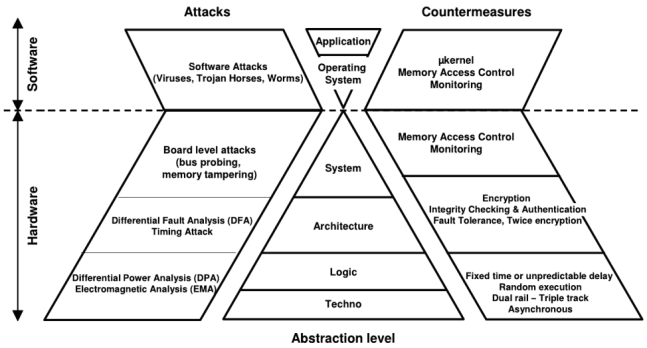

Figure 15. Hardware threats and countermeasures by abstraction level. Source: [69]

For instance, when implementing security modules in FPGA, we are working at the Architecture and Logic levels, which are vulnerable to Side Channel Attacks, such as Differential Power Analysis [64] and Timing Attacks [63]. These threats are called SCA because rather than attack cryptographic algorithms vulnerabilities, the attackers observe "leaked"information of the cryptographic circuit such as power consumption or the amount of time to compute one specific cryptographic function.

So we must worry about integrity checking, bitstream encryption, fault tolerance, and some low level countermeasures

such as unpredictable random (or fixed time) execution, dual-rail or asynchronous implementation. However, if carefully designed, cryptographic modules in FPGA can be efficient (in terms of performance, cost and flexibility) and secure.

In [70] and [71] we demonstrate the efficiency of FPGA to implement Montgmomery Modular Multiplications [72], essential to public key cryptosystems such as RSA. In [73] we show how to counteract DPA attacks with Reconfigurable Computing using a leak resistant arithmetic and architecture. Fig. 16 shows an ovewview of this approach.

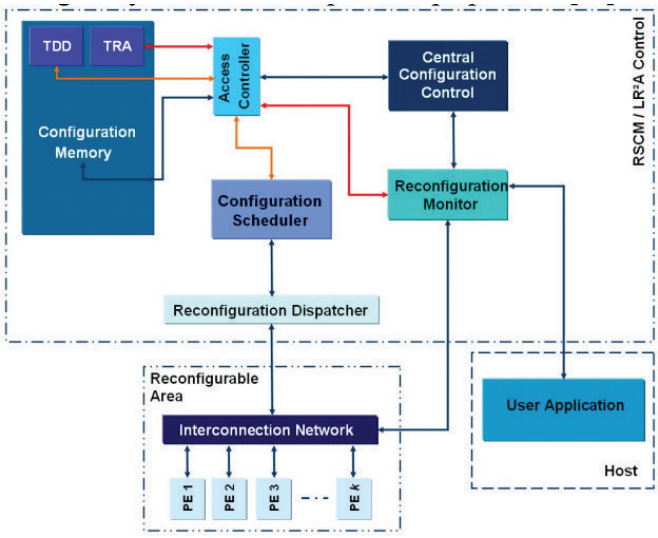

Figure 16. A leak resistant reconfigurable architecture. Source: [73]

The "user application"sends a request to the reconfigurable controller module in order to specify its needs in terms of cryptography. The reconfigurable monitor talks to the central configuration control to verify if the requested function is available in the configuration memory. If so, the configuration scheduler searches the corresponding cryptographic function in the memory and dispatches it to the reconfigurable area, in order to be executed. On the example, the cryptographic module is a Leak Resistant Arithmetic to perform Montgomery modular multiplications capable of masking power consumption and computing time so no DPA or Differential Fault Analysis (DFA) can be performed against the circuit.

These few works illustrates the capabilities of FPGA to cope with network cryptographic needs.

D. Section Summary

The examples in this section shows that Reconfigurable Hardware is already in use for both network and security applications, being useful for rapid hardware prototyping but also as market solution on these fields. However, the Future Internet initiatives do not mention explicitly the use of Reconfigurable Computing in its directive, nor its related projects give attention to this possibility.

IV. Conclusions

In this article we try to logically build the argument that it is necessary to approximate the areas of computer architecture and computer networks, or more specifically bridge the gap between research in Reconfigurable Computing those in the Future Internet Architectures. We show through a brief survey full of successful examples, such as some of the needs and future internet objectives can be met through the reconfigurable computing.

Finally, some considerations on lessons we learned in building this article.

First, we assume that to approach the research of Future Internet and Reconfigurable Computing, need to adopt an approach that, in parallel, is top-down and bottom-up.

On one hand we think the top-down approach in the sense that researchers in both areas perform specific meeting to discuss joint research. It is also necessary to articulate actions so that there is the funding for such initiatives, preferably through public-private partnerships.

On the other hand, the bottom-up approach means that complexity of current research and development has no room for compartmentalized knowledge. The undergraduate curricula should emphasize pedagogical approaches based on multidisciplinary problem solving [74], [75]. After all, accustoming students to think in a complex way, we increase the chances of developing the critical mass necessary for troubleshooting problems composed of variable increasingly numerous and diverse in nature. According to Edgar Morin[76]:

> *"We need a kind of thinking that reconnects that which is disjointed and compartmentalized, that respects diversity as it recognizes unity, and that tries to discern interdependencies. We need a radical thinking (which gets to the root of problems), a multidimensional thinking, and an organizational or systemic thinking."*

As shown in the examples brought previously, the merge of fields Reconfigurable Computing and Future Internet is feasible. However this integration needs to be intensified and adopted as early as possible so that, from the specification phase of the solutions for the Internet of the Future, already consider Reconfigurable Computing as architectural alternative for implementation.

Acknowledgments

The authors would like to thank the Brazilian CAPES agency for support the project "Rede de Cooperação Universitária para Ensino Superior e Pesquisa Avançada Científica e Tecnológica em Sistemas Eletrônicos e Interativos aplicados à Defesa Nacional"CAPES-PRODEFESA, CAPES-PRODEFESA number 23038.009307/2013-77.

References

[1] J. Postel, "Internet protocol," Internet Requests for Comments, RFC Editor, RFC 791, September 1981. [Online]. Available: https://tools.ietf.org/html/rfc791

[2] C. S. University. (2016) Figure of web. [Online]. Available: http://som.csudh.edu/cis/lpress/history/arpamaps/press6.jpg

[3] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sept 1995.

[4] CIA. (2016) Country comparison: Internet hosts. [Online]. Available: https://www.cia.gov/library/publications/the-world-factbook/rankorder/2184rank.html

[5] T. O. Project. (2016) Figure of web. [Online]. Available: http://www.opte.org/the-internet/

[6] M. Sailan, R. Hassan, and A. Patel, "A comparative review of ipv4 and ipv6 for research test bed," in *2009 International Conference on Electrical Engineering and Informatics*, vol. 02, Aug 2009, pp. 427–433.

[7] S. Frankel and S. Krishnan, "Ip security (ipsec) and internet key exchange (ike) document roadmap," Internet Requests for Comments, RFC Editor, RFC 6071, February 2011. [Online]. Available: https://tools.ietf.org/html/rfc6071

[8] G. Estrin, "Organization of computer systems: The fixed plus variable structure computer," in *Papers Presented at the May 3-5, 1960, Western Joint IRE-AIEE-ACM Computer Conference*, ser. IRE-AIEE-ACM '60 (Western). New York, NY, USA: ACM, 1960, pp. 33–40. [Online]. Available: http://doi.acm.org/10.1145/1460361.1460365

[9] J. Rose, A. E. Gamal, and A. Sangiovanni-Vincentelli, "Architecture of field-programmable gate arrays," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 1013–1029, Jul 1993.

[10] G. Estrin, "Reconfigurable computer origins: the ucla fixed-plus-variable (f+v) structure computer," *IEEE Annals of the History of Computing*, vol. 24, no. 4, pp. 3–9, Oct 2002.

[11] Digilent. (2016) Netfpga sume virtex 7 development board. [Online]. Available: goo.gl/0b4C6P

[12] Terasic. (2016) De5net fpga development kit. [Online]. Available: goo.gl/o3pWNM

[13] M. Boniface, M. Calisti, and M. Serrano, "Next generation internet experimentation." European Comission, Tech. Rep., June 2016. [Online]. Available: https://goo.gl/kZq0VG

[14] RNP. (2016) Internet do futuro. [Online]. Available: https://goo.gl/w8JjkD

[15] C. Rothenberg, A. Vidal, M. Salvador *et al.*, "Hybrid networking toward a software-defined era," in *Network Innovation through OpenFlow and SDN*, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. Oxford: CRC Press, 2014, ch. 8, pp. 153–198.

[16] L. Ciuffo, T. Salmito, J. Rezende, and I. Machado, "Testbed fibre: Passado, presente e perspectivas," in *Anais do Workshop de Pesquisa Experimental da Internet do Futuro*, vol. 1. SBC, Jun 2016, pp. 3–7.

[17] FIBRE. (2016) Figure of web. [Online]. Available: http://fibre.org.br/infrastructure/resources/

[18] NSF. (2016) National science foundation future internet architecture project. [Online]. Available: http://www.nets-fia.net/

[19] L. Zhang, E. Estrin, J. Burke *et al.*, "Named data networking (ndn) project," NDN Project, Tech. Rep., October 2010. [Online]. Available: http://goo.gl/K9YsV8

[20] M. Murata, "Goals of r&d on new-generation network project," *Journal of the National Institute of Information and Communications Technology*, vol. 62, no. 2, pp. 2–5, Mar 2015.

[21] N. Nishinaga and H. Harai, "Progress and results of new-generation network research and development," *Journal of National Institute of Information and Communications Technology*, vol. 62, no. 2, pp. 1176–1188, March 2016.

[22] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, July 2011.

[23] N. D. Networking. (2016) Figure of web. [Online]. Available: https://named-data.net/project/execsummary/

[24] S. Hauck, "The roles of fpgas in reprogrammable systems," *Proceedings of the IEEE*, vol. 86, no. 4, pp. 615–638, Apr 1998.

[25] R. Hartenstein, "A decade of reconfigurable computing: A visionary retrospective," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '01. Piscataway, NJ, USA: IEEE Press, 2001, pp. 642–649. [Online]. Available: http://dl.acm.org/citation.cfm?id=367072.367839

[26] R. Tessier, K. Pocek, and A. DeHon, "Reconfigurable computing architectures," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 332–354, March 2015.

[27] S. Brown and J. Rose, "Fpga and cpld architectures: a tutorial," *IEEE Design Test of Computers*, vol. 13, no. 2, pp. 42–57, Summer 1996.

[28] K. Compton and S. Hauck, "Reconfigurable computing: A survey of systems and software," *ACM Computing Surveys*, vol. 34, no. 2, pp. 171–210, Jun. 2002. [Online]. Available: http://doi.acm.org/10.1145/508352.508353

[29] C. Maxfield, *The Design Warriorś Guide tho FPGAs: Devices, Tools and Flows*. Massachusetts, USA: Elsevier, 2004.

[30] K. Compton and S. Hauck, "An introduction to reconfigurable computing," Northwestern University, Tech. Rep., October 1999. [Online]. Available: http://goo.gl/gJgIEv

[31] XILINX. (2016) Figure of web. [Online]. Available: http://www.xilinx.com/fpga/asic.htm

[32] TERASIC. (2016) Figure of web. [Online]. Available: http://www.terasic.com

[33] O. N. Fondation. (2016) Software-defined networking (sdn) definition. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition

[34] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, July 2013.

[35] A. Kalyaev and E. Melnik, "Fpga-based approach for organization of sdn switch," in *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*, Oct 2015, pp. 363–366.

[36] N. Zilberman, Y. Audzevich, G. Kalogeridou *et al.*, "Netfpga: Rapid prototyping of networking devices in open source," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: ACM, 2015, pp. 363–364. [Online]. Available: http://doi.acm.org/10.1145/2785956.2790029

[37] G. Antichi, M. Shahbaz, Y. Geng *et al.*, "Osnt: open source network tester," *IEEE Network*, vol. 28, no. 5, pp. 6–12, September 2014.

[38] Y. E. Kwasi and R. Rojas-Cessa, "High-resolution hardware-based packet capture with higher-layer pass-through on netfpga card," in *2014 23rd Wireless and Optical Communication Conference (WOCC)*, May 2014, pp. 1–6.

[39] G. Ibáñez, J. A. Carral, E. Rojas, and J. M. Giménez-Guzmán, "Evaluating native load distribution of arp-path bridging protocol in mesh and data center," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3769–3774.

[40] X. Liu, A. Wada, T. Xing, P. Juluri, Y. Sato, S. Ata, D. Huang, and D. Medhi, "Servitr: A framework for trust and policy management for a secure internet and its proof-of-concept implementation," in *2012 IEEE Network Operations and Management Symposium*, April 2012, pp. 1159–1166.

[41] D. Mesquita, R. Oliveira, and P. Rosa, "Harp - high availability routing protocol," 2014, bR Patent BR5120130001056.

[42] J. H. de Souza Pereira, F. de Oliveira Silva, E. L. Filho, S. T. Kofuji, and P. F. Rosa, *Title Model Ontology for Future Internet Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 103–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20898-0_8

[43] G. Hashimoto, E. Filho, J. Pereira, and P. Rosa, "High avaliability: A long-term feature in network elements," in *Proceedings of the Fifth International Conference on Systems and Networks Communications (ICSNC), 2010*, aug. 2010, pp. 201 –206.

[44] R. Oliveira, D. Mesquita, and P. Rosa, "Harp: A split brain free protocol implemented in fpga," in *Proceedings of the Ninth Advanced International Conference on Telecommunications, AICT 2013*, vol. 9, jun 2013, pp. 197–203. [Online]. Available: https://www.thinkmind.org/download.php?articleid=aict_2013_9_20_10173

[45] H. Chen, Y. Chen, and D. H. Summerville, "A survey on the application of fpgas for network infrastructure security," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 541–561, 2011.

[46] J. Li, Y. Chen, C. Ho, and Z. Lu, "Binary-tree-based high speed packet classification system on fpga," in *The International Conference on Information Networking 2013 (ICOIN)*, Jan 2013, pp. 517–522.

[47] Y. R. Qu and V. K. Prasanna, "High-performance and dynamically updatable packet classification engine on fpga," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 197–209, Jan 2016.

[48] A. Wicaksana and A. Sasongko, "Fast and reconfigurable packet classification engine in fpga-based firewall," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, July 2011, pp. 1–6.

[49] Y. R. Qu, H. H. Zhang, S. Zhou, and V. K. Prasanna, "Optimizing many-field packet classification on fpga, multi-core general purpose processor, and gpu," in *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*, May 2015, pp. 87–98.

[50] K. Jaic, M. C. Smith, and N. Sarma, "A practical network intrusion detection system for inline fpgas on 10gbe network adapters," in *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, June 2014, pp. 180–181.

[51] W.-S. Jung and T. G. Kwon, "An independently partial pattern matching for content inspection at multi gigabit networks," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2, Feb 2010, pp. 1574–1579.

[52] T. T. Hieu and N. T. Tran, "A memory efficient fpga-based pattern matching engine for stateful nids," in *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2013, pp. 252–257.
[53] C. Xu, S. Chen, J. Su, S. M. Yiu, and L. C. K. Hui, "A survey on regular expression matching for deep packet inspection: Applications, algorithms and hardware platforms," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.
[54] M. Cai, K. Hwang, Y.-K. Kwok, S. Song, and Y. Chen, "Collaborative internet worm containment," *IEEE Security Privacy*, vol. 3, no. 3, pp. 25–33, May 2005.
[55] J. W. Lockwood, J. Moscola, M. Kulig, D. Reddick, and T. Brooks, "Internet worm and virus protection in dynamically reconfigurable hardware," in *in '03 Military and Aerospace Programmable Logic Device (MAPLD)*, Washington, DC, Sep 2003, p. E10.
[56] K. Pandiyarajan, S. Haridas, and K. Varghese, "Transparent fpga based device for sql ddos mitigation," in *Field-Programmable Technology (FPT), 2013 International Conference on*, Dec 2013, pp. 82–89.
[57] J.-T. Oh, S.-K. Park, J.-S. Jang, and Y.-H. Jeon, "Detection of ddos and ids evasion attacks in a high-speed networks environment," *International Journal of Computer Science and Network Security*, vol. 7, no. 6, pp. 124–131, Jun 2007. [Online]. Available: http://paper.ijcsns.org/07_book/200706/20070617.pdf
[58] W. N. Chelton and M. Benaissa, "Fast elliptic curve cryptography on fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 2, pp. 198–205, Feb 2008.
[59] N. Nedjah and L. M. Mourelle, "Three hardware architectures for the binary modular exponentiation: sequential, parallel, and systolic," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 3, pp. 627–633, March 2006.
[60] H. W. Kim and S. Lee, "Design and implementation of a private and public key crypto processor and its application to a security system," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 214–224, Feb 2004.
[61] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: http://doi.acm.org/10.1145/359340.359342
[62] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computations*, vol. 48, pp. 203–209, 1987.
[63] P. C. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.
[64] P. Kocher, J. Jaffe, and B. Jun, *Differential Power Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.
[65] ALTERA. (2016) Anti-tamper capabilities in fpga designs. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01066-anti-tamper-capabilities-fpga.pdf
[66] ——. (2016) Developing tamper resistant designs with xilinx virtex-6 and 7 series fpgas. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp1084_tamp_resist_dsgns.pdf
[67] A. Seffrin, S. Malipatlolla, and S. A. Huss, "A novel design flow for tamper-resistant self-healing properties of fpga devices without configuration readback capability," in *Field-Programmable Technology (FPT), 2010 International Conference on*, Dec 2010, pp. 291–294.
[68] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 760–768, June 2004.
[69] B. Badrignans, J.-L. Danger, V. Fischer, G. Gognat, and L. Torres, *Security Trends on FPGA*. Montpellier, France: Springer-Verlag, 2011.
[70] D. G. Mesquita, G. Perin, F. L. Herrmann, and J. a. B. d. S. Martins, "An efficient implementation of montgomery powering ladder in reconfigurable hardware," in *Proceedings of the 23rd Symposium on Integrated Circuits and System Design*, ser. SBCCI '10. New York, NY, USA: ACM, 2010, pp. 121–126. [Online]. Available: http://doi.acm.org/10.1145/1854153.1854184
[71] D. G. Mesquita, G. Perin, and J. a. B. d. S. Martins, "Montgomery modular multiplication on reconfigurable hardware: Systolic versus multiplexed implementation," *International Journal of Reconfigurable Computing*, vol. 2011, pp. 1–10, January 2011. [Online]. Available: http://dx.doi.org/10.1155/2011/127147
[72] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of Computations*, vol. 44, no. 170, pp. 519–521, 1985.
[73] D. Mesquita, B. Badrignans, L. Torres, G. Sassatelli, M. Robert, and F. Moraes, "A cryptographic coarse grain reconfigurable architecture robust against dpa," in *2007 IEEE International Parallel and Distributed Processing Symposium*, March 2007, pp. 1–8.
[74] N. Linge and D. Parsons, "Problem-based learning as an effective tool for teaching computer network design," *IEEE Transactions on Education*, vol. 49, no. 1, pp. 5–10, Feb 2006.
[75] J. H. Lee, S. E. Lee, H. C. Yu, and T. Suh, "Pipelined cpu design with fpga in teaching computer architecture," *IEEE Transactions on Education*, vol. 55, no. 3, pp. 341–348, Aug 2012.
[76] E. Morin, *On Complexity*. Hampton Press, 2008.

Daniel Mesquita is associate professor at the Unipampa (Federal University of the Pampa, Brazil). He received his Ph.D. degree in Microelectronics from Université Montpellier II (France) for his thesis on "Reconfigurable architectures and cryptography". In 2007 and 2008 he worked as researcher at the Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID), in Lisbon, Portugal. In 2008 and 2009 Daniel worked as developer at the CEITEC S.A., a Brazilian semiconductor company. Since 1999 Daniel discusses reconfigurable computing trends, tools and applications. His current research concerns the use of reconfigurable computing to improve security, reliability and flexibility to future internet.

Pedro Frosi is titular professor at the UFU (Federal University of Uberlandia, Brazil). He received his Ph.D. degree in Computer Engineering from the University of Sao Paulo (USP), Brazil and the Centre National pour la Recherche Scientifique, France, on the field of distributed systems. He holds a master degree on computer architecture from USP. He's research interest concern Future Internet, Internet of Things, Cloud Computing, High Availability Architectures and Scalability for Software Architecture.