Applied Soft Computing 12 (2012) 1440–1450

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Enabling fuzzy technologies in high performance networking via an open FPGA-based development platform

Federico Montesino Pouzols[a,d,*], Angel Barriga Barros[b], Diego R. Lopez[c], Santiago Sánchez-Solano[d]

[a] Department of Biosciences, Center of Excellence in Metapopulation Biology, University of Helsinki, Viikinkaari 1, P.O. Box 65, FI-00014, Finland
[b] Department of Electronics and Electromagnetism, University of Seville, Avda. Reina Mercedes s/n. Edif. CICA, E-41012 Seville, Spain
[c] Telefónica I+D, Don Ramón de la Cruz, 82-84, E-28006 Madrid, Spain
[d] Microelectronics Institute of Seville (IMSE-CNM), CSIC, C. Americo Vespucio s/n, E-41092 Seville, Spain

ARTICLE INFO

Article history:
Received 3 April 2009
Received in revised form 29 July 2011
Accepted 23 October 2011
Available online 12 November 2011

Keywords:
Fuzzy inference
Field programmable gate arrays
Computer networks
Network traffic control
Congestion control
Queuing control
Network performance

ABSTRACT

Soft computing techniques and particularly fuzzy inference systems are gaining momentum as tools for network traffic modeling, analysis and control. Efficient hardware implementations of these techniques that can achieve real-time operation in high-speed networking equipment as well as other highly time-constrained application fields is however an open problem. We introduce a development platform for fuzzy inference systems with applications to network traffic analysis and control. The platform addresses the current requirements and constraints of high performance networking equipment. For the development process, we set up a methodology and a CAD tool chain that span the entire design process from initial specification in a high-level language to implementation on FPGA devices. An FPGA development board with PCI/PCIe interface is employed to support an open platform that comprises CAD tools as well as IP cores. PCI compatible fuzzy inference modules are implemented as System-on-Programmable-Chip (SoPC). We present satisfactory experimental results from the implementation of fuzzy systems for a number of applications in analysis and control of Internet traffic. These systems are shown to satisfy operational and architectural requirements of current and future high performance routing equipment. The platform proposed allows for the development of prototypes while avoiding large investments and complicated management procedures which constrain the testing and adoption of soft computing techniques in high performance networking.

1. Introduction

Soft computing techniques and particularly fuzzy inference systems have been gaining momentum during the last decade as tools for modeling, analysis and control of network traffic. Fuzzy inference systems have been shown to be effective in areas such as traffic control in routers [16,28,76,68,31,13], support for differentiated services within the DiffServ architecture [58,77,74,18], real-time traffic measurement, analysis and monitoring [61,63,62,64], power and quality of service optimization for wireless networks [55,72], as well as end-to-end traffic control [65,27] and buffer control [49,64]. Efficient hardware implementations of these techniques that can achieve real-time operation in high-speed communications equipment as well as many other demanding application fields is however an open problem.

Current routing architectures pose two major challenges in the design of new mechanisms: *scalability* and *flexibility* of implementations. Here we introduce a platform and a companion development methodology for developing fuzzy systems that does not only fulfill operational requirements but also addresses the challenges posed by current routing architectures.

A major research problem in Internet transport and network layers is the development of traffic regulation mechanisms that can cope with the requirements of a growing diversity of technologies, applications and services. More generally, Internet traffic dynamics is an increasingly complex topic of research [66].

As stated above, diverse research results show that fuzzy inference systems can help solve current problems in Internet traffic control. However, while many industrial applications of fuzzy systems in a variety of fields have been reported, fuzzy systems for traffic control have not yet found their way into real-world applications. Despite the good performance of the aforementioned fuzzy logic based mechanisms for traffic analysis and control, there is a lack of architectures and design procedures for implementing

* Corresponding author at: Microelectronics Institute of Seville (IMSE-CNM), CSIC, C. Americo Vespucio s/n, E-41092 Seville, Spain. Tel.: +34 954 466 666; fax: +34 954 466 600.
E-mail addresses: federico.montesinopouzols@helsinki.fi (F.M. Pouzols), barriga@us.es (A.B. Barros), diego@tid.es (D.R. Lopez), santiago@imse-cnm.csic.es (S. Sánchez-Solano).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.10.018

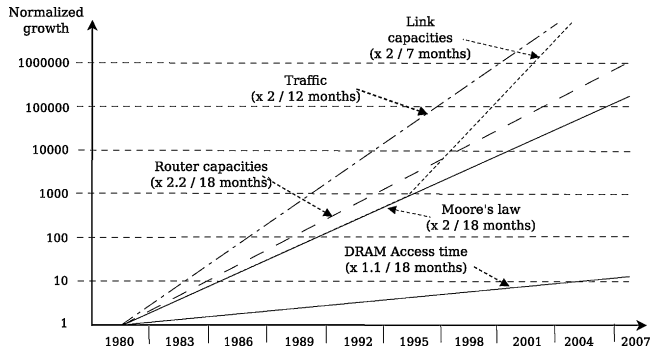

Fig. 1. High-end routers technological trends (log scale) [38,39,47,19,46,15,56].

them in a systematic manner yet addressing current challenges in high performance networking systems. As a consequence, although significant results in diverse applications of fuzzy systems in communications and networking have been reported since more than a decade [35], the deployment of these systems in the real world is still a challenge.

Both academia and major vendors are currently pushing for distributed and modular router designs, specially for high-end backbone routers. In these designs routers are composed of modules that can be mapped onto different processing elements which communicate over an internal network [38]. Hardware for high-end networking systems has been traditionally developed in a custom and unstructured manner. Nevertheless, reconfigurable architectures are employed in practice by most vendors and design methodologies for easing the development process are sought.

Technological trends in Internet core routers and high-end communications hardware in general (see Fig. 1) lead to hard constraints specially regarding packet processing rates. During the last years total Internet traffic has grown at over 80 percent per year, which directly translates into a similar or even higher increase of traffic volume in backbone routers. Overall, network traffic volume increases at a rate that outpaces advances in VLSI technology. Within this context, two main constraints arise: scalability (processing units must be able to process up to millions of packets per second (Mpps or Mp/s)), and flexibility and reconfigurability of implementations (a requirement imposed by the fast increasing diversity of protocols and technologies involved) [38,70].

As further discussed in the next section, a number of technological and architectural factors impose strict constraints on traffic analysis and control mechanisms in high performance networking equipment. Hence, efficient yet flexible hardware implementations of soft computing methods and specially fuzzy inference systems that can achieve real-time operation in high-speed networking equipment are needed. The motivation for this work is to overcome these practical difficulties. The major contributions of this paper are as follows. The implementation of fuzzy inference systems is made possible within the practical constraints of contemporary high performance networking equipment. To this end, we provide a development platform which includes a complete methodology and automated design flow, as well as CAD tools. In addition, these systems are evaluated in simulated, emulated and test scenarios. Finally, this paper reports prototypes that provide evidence for the feasibility of these soft computing systems in contemporary high performance networking equipment.

The next section outlines current implementation issues in networking equipment which motivate this work. Then, Section 3 describes the development platform architecture defined for implementing fuzzy systems. Section 4 deals with the development methodology and design flow applied. Simulation and experimental implementation results are then presented in Section 5. Hardware (FPGA-based) implementation results are described and discussed in Section 6. Finally, we conclude in Section 7.

2. Implementation constraints in networking equipment

Routers can be classified into three classes depending on the level of deployment within the Internet. These classes correspond to access routers, campus or enterprise routers and core routers. The focus of this paper is particularly on core or high-end routers, those designed for network backbones. By tying together networks of the global Internet, routers made up a unified whole. While the main function of a router is to forward packets from a set of input links to a set of output links, they must also implement complex distributed routing algorithms, deal with diverse link technologies and provide support for traffic engineering tasks, differentiated services as well as quickly evolving quality of service schemes.

The architecture of Internet routers has evolved at a fast rate since the first implementations appeared [36,67,19,46]. Drastic changes have taken place during the past two decades [14,22,24,48,6,70,15]. This evolution has been driven by a number of technological trends and functional requirements. On the one hand, the divergence in performance increase seen by the diverse components of a router (such as memory elements, interconnection links, programmable devices and processors, see Fig. 1) challenges router design. On the other hand, new functional requirements have arised as new applications, services and technologies are being deployed. As a consequence, a great deal of research efforts is ongoing to address these challenges in router design.

In this context, high throughput IP routers require that critical tasks be identified and isolated, using tailored special purpose modules to perform them [6]. The basic principle in routing hardware design and Network Processing Unit (NPU) based systems [23,26,25,45,24] is to exploit parallelism against the main limiting factor imposed on overall performance: the memory access speed. There are two main alternatives for the hardware implementation of an NPU: architectures based on general purpose processors, such as the NPUs from a number of vendors [39], and specific architectures, with better performance but lower flexibility. In the latter case, the development time is excessively long as compared to the fast evolution of applications, protocols and services.

Presently, a hybrid approach is followed to satisfy flexibility requirements. This leads to designs where high-end routers use general purpose units together with co-processing and acceleration elements (processing engines) implemented as ASIC or on specific FPGA devices. FPGA devices are thus used not only for developing prototypes but also for final products. In the current Internet, link speeds and packet processing rates are quickly increasing. The pace at which the speed of memory elements as well as other components of the router processing units increases is significantly slower. That is, the rates at which two key technological factors evolve have been increasingly diverging, and it is expected that this trend will continue. As a consequence, many data storage and processing elements (or processing engines) in current routers are implemented by means of specialized processing engines using specific hardware architectures [24–26,38,23].

Regarding traffic measurement, analysis and control, specific hardware architectures are sought in order to cope with the increasing packet processing rates. Specific hardware architectures have been proposed for measurement [30], analysis of network flows [75] and implementing a number of common processing engines [24–26]. In particular, when implementing active queue management schemes, the packet queue control rate must be very close to the maximum packet processing speed attainable by a