FPGA BASED IMPLEMENTATION OF FUZZY CONTROLLERS FOR INTERNET TRAFFIC

Federico Montesino Pouzols[1], *Ángel Barriga Barros*[1], *Diego R. Lopez*[2], *Santiago Sánchez-Solano*[1]

[1]Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC
[2]RedIRIS, Red Académica y de Investigación Española

Federico.Montesino@imse.cnm.es, Angel.Barriga@imse.cnm.es, Diego.Lopez@rediris.es,
Santiago.Sanchez@imse.cnm.es

ABSTRACT

Recent research results propose and show the usefulness of a fuzzy control based approach to the development of intelligent systems for congestion control in Internet routers. However, adoption of this new technology is handicapped because of operational requirements, mostly in terms of inference speed, a hard constrain on the practical implementation of key traffic controlling systems. We report on the implementation of intelligent fuzzy controllers for Internet traffic using an FPGA based prototyping platform. A development methodology and a tool chain, a flexible and open prototyping platform, a set of fuzzy controllers and implementation results for a number of traffic controllers are presented. Our prototypes are shown to satisfy the requirements of high performance routing hardware deployed in the current Internet.

1. INTRODUCTION

The development of intelligent systems for congestion control (and traffic control in general) in the Internet is an open research area. Recent results propose and show the usefulness of a fuzzy control based approach [3,23,20,18,8,21].

However, operational requirements of Internet traffic controlling systems, mostly in terms of processing speed, impose a hard constrain on its practical implementation. Thus, a limiting factor for the real-world deployment of fuzzy controllers for Internet traffic is the need for efficient implementations that can achieve the high inference rates required by current and future high performance networks. Microelectronic implementations with specific and efficient architectures must be employed in this application area of fuzzy logic based systems, specially in the case of current and future high performance network backbones [14].

More generally, Internet traffic dynamics remains an open topic of research. Soft computing systems can provide intelligent solutions for both traffic analysis and control provided they can operate in real-time in current high speed networks. Thus, the availability of efficient microelectronic implementations has two important implications:

- Opens a new application area of fuzzy systems in Internet.
- Fosters the research on fuzzy logic based solutions to Internet traffic control and analysis.

We report on the implementation of intelligent fuzzy controllers for Internet traffic using an FPGA based prototyping platform. Prototype implementation results are shown to satisfy the requirements of high performance routing hardware deployed in the current Internet.

In section 2, we present an open prototyping platform defined for easing the development and validation of Internet traffic controllers. Section 3 describes a companion development methodology and tool chain that covers the full design flow from specification to prototype implementation and validation. A representative subset of the traffic controllers we have implemented is outlined in section 4. In section 5, prototypes hardware implementation results are given. Finally, we summarize conclusions from our work and future steps.

2. PROTOTYPING PLATFORM

Throughout more than a decade, strategies and methodologies for prototyping fuzzy logic based controllers have been developed. To date, most work on this topic has been focused on industrial applications [2]. Considering the specific requirements as well as the high cost and complexity of high performance routers currently deployed on the Internet, we have developed a flexible prototyping platform for Internet traffic controllers. This platform has been defined with a twofold objective:

- Easing the microelectronic implementation of prototypes of Internet traffic controllers.
- Providing a complete set of tools and environment for realistic validation.

Our prototype development architecture, depicted in figure 1, is based on a conventional PC equipped with an FPGA development board with PCI interface, thus

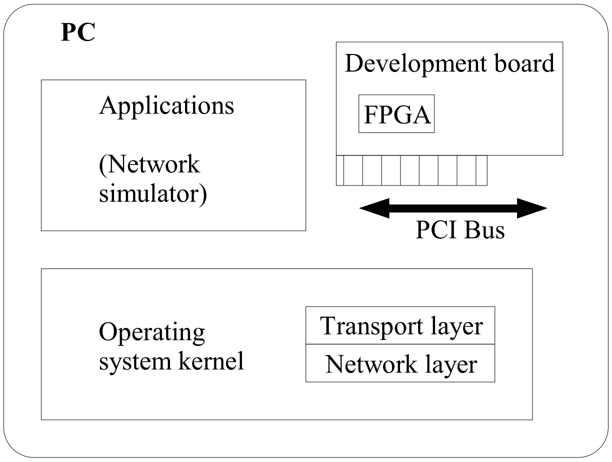


Figure 1: Prototyping platform scheme.

making a flexible and cheap solution with no specific hardware requirements yet able to emulate the behavior of complex and expensive routing equipment. As shown in figure 2, when implementing fuzzy systems, we consider two main function blocks: those directly related to fuzzy inference and those that can be classified as auxiliary functions (such as initialization, timing, pre/post-processing, etc.) [2]. For the implementation of prototypes of fuzzy controllers for Internet traffic the following model is used:

- A fuzzy inference module (FIM) is implemented on hardware described by means of VHDL according to an specific implementation architecture tailored for efficient and fast fuzzy inference. The methodology and tools employed for the development of the FIM is described in the next section.
- All auxiliary functions are implemented as software. Software can run on the PC operating system as well as on optional components implemented on the FPGA of the development board.

A flexible and open architecture for implementing fuzzy systems on the FPGA has been defined. Within this architecture, depicted in figure 3, FIM modules are integrated as subsystems of a potentially complex and reconfigurable fuzzy logic based digital system.

Interconnection between the fuzzy digital system and the host PC is done through a standard PCI bus. The internal bus of the fuzzy digital system is a WISHBONE [9] bus , a SoC interconnection architecture for portable IP cores, that connects a variable number of components :

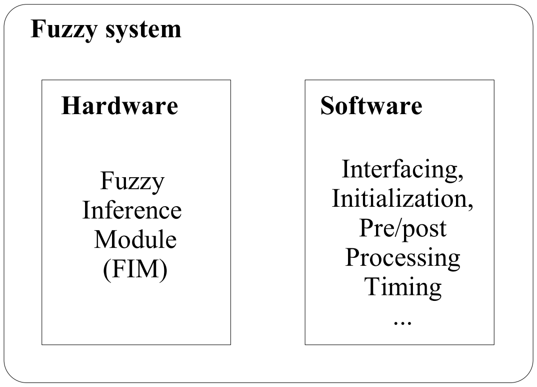

Fuzzy system
Hardware
Fuzzy Inference Module (FIM)
Software
Interfacing, Initialization, Pre/post Processing Timing ...

Figure 2: Hardware and software tasks.

- FIM.
- PCI-WISHBONE bridge and WISHBONE bus controller.
- Additional cores (Network Interface Card, Processor , etc.).

We are currently using an AvNet ADS-XLX-SP3-EVL1500 FPGA development board. Both the WISHBONE [9] bus and the PCI-WISHBONE Bridge [6] IP cores have been developed under free distribution licenses by the OpenCores [17] organization as well as other entities. WISHBONE Systems can easily interface with OPB [19] based systems through the WISHBONE-OPB bridge provided as a plug-in for Xilinx EDK and also available under a free distribution license.

This way, software tasks can be defined using common programming languages and can be run on the generic purpose processing units of the PC as well as on specific processing units included in the FPGA. For instance, a fuzzy logic based traffic analysis application can be implemented incorporating an OpenRisc Core for which the GNU/Linux operating system is available.

The PCI interface of the prototypes eases integration with routing architectures by major vendors. Within routing architectures currently deployed in the Internet [10], the fuzzy controllers could be seamlessly integrated as processing engines whether at the NPU and/or the output/input cards depending on the quality of service architecture implemented on the router.

In addition, when the development board employed includes a network interface card, as is the case with our AvNet board, a whole fuzzy logic based traffic analysis application can be implemented as a standalone SoPC on the FPGA.

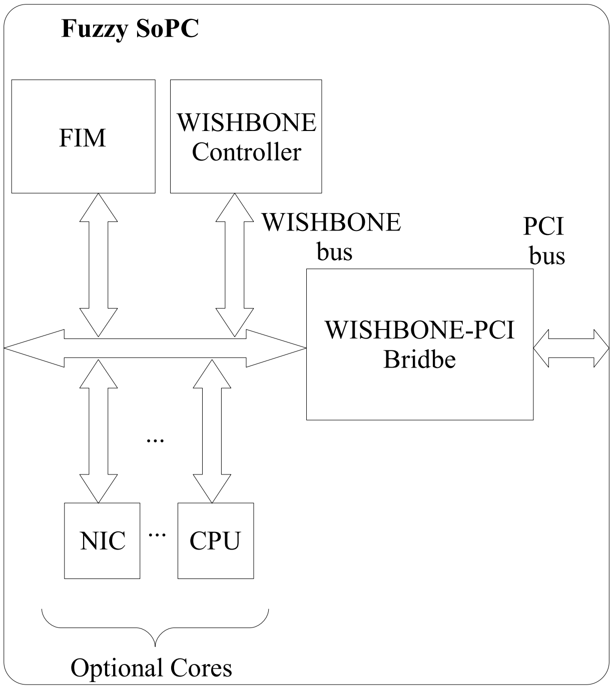


Figure 3: Fuzzy systems prototyping architecture.

3. DEVELOPMENT METHODOLOGY AND TOOL CHAIN

As a result of more than 10 years of research experience on the digital implementation of fuzzy systems, the fuzzy group at IMSE has developed methodologies and CAD tools that fulfill the design flow of fuzzy systems. We leverage on the Xfuzzy [15] CAD suite of tools and a methodology [2] for the development of fuzzy controllers to define a methodology and tool chain tailored for the development of fuzzy Internet traffic controllers.

The design flow and tool chain we have defined and used to develop fuzzy inference modules is depicted in figure 4.

The design flow covers the whole development process, from initial specification to final implementation whether as software or hardware. The first development stage (description) is performed using a high level fuzzy systems specification language, XFL [16], which can later be turned into C and VHDL code among other implementation options.

The tool chain includes:

- The xfc and xfcpp tools (included in Xfuzzy), which turns an XFL specification into C and C++ code.
- The xfvhdl tool (included in Xfuzzy), which turns an XFL specification into VHDL code generated for a specific efficient architecture for the implementation of fuzzy systems [15].
- ns-2 [11], an open network simulator widely spread within the Internet research community.
- Operating system kernel (currently Linux and FreeBSD).

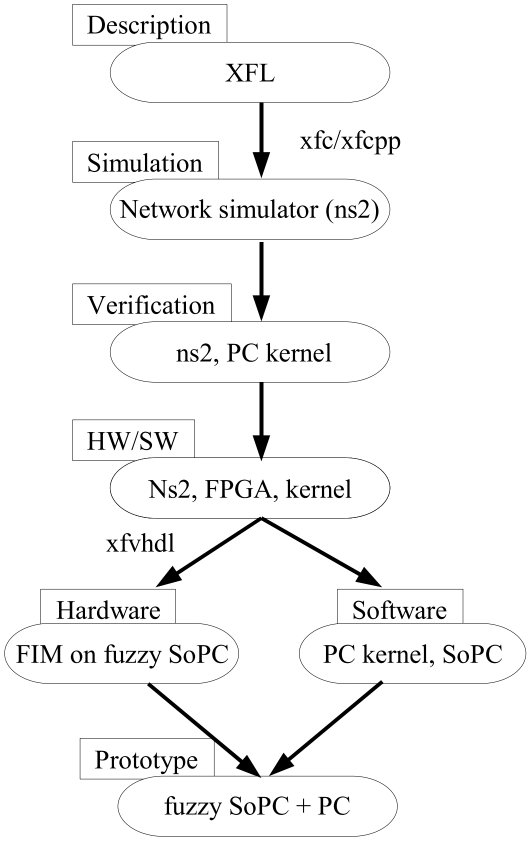


Figure 4: Design flow and tool chain.

The development stages after specification have been tailored for Internet traffic controller development as follows.

For network simulation, we have used ns-2. ns-2 is an object oriented discrete event driven simulator with support for a vast variety of transport protocols, queueing systems, routing schemes and access media, thus enabling us to evaluate the performance of traffic controllers under complex and realistic simulated scenarios. Fuzzy controllers are integrated into ns-2 as components implemented in C.

Verification can be performed over software and hardware implementations of fuzzy controllers. Software verification is performed over a controller implementation within the kernel of the general purpose operating system of the PC. For verifying hardware prototypes of FIMs, two procedures have been considered:

- Verification by means of network simulators. To this end, code and drivers to make it possible to access the FIM in the FPGA development card from ns-2 has been developed.
- Verification through emulated scenarios where a router is emulated by means of the prototyping platform. Validation in real or emulated scenarios is also possible with our prototyping architecture by using the prototyping PC as a router. This accomplished by replacing queue control functionality in the operating system network layer with functionality provided by the FIM in the development board. To this end, kernel drivers have been developed to make it possible to access the fuzzy controller in the FPGA development card from networking modules in the operating system kernel. Drivers have been developed for FreeBSD and Linux kernels.

We have defined as general hardware-software partition the implementation on hardware of the FIM module whereas all other tasks are implemented as software.

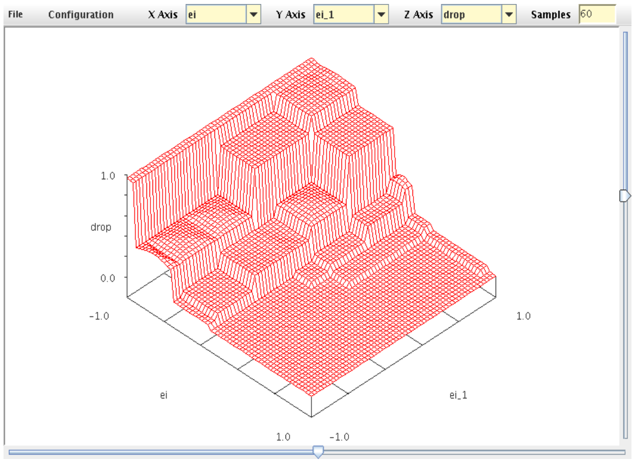
Implementation of novel hardware components and experimental deployment on high-end equipment poses major practical problems. Deployment on high-end (around 1 million euro cost per unit) routing equipment requires the adoption of a new technology by vendors of routing hardware (a market with high inertia), which is a long term objective of our research. Nonetheless, by means of our prototype architecture, validation can be performed the same way as verification through emulated scenarios as described above.

By following a well defined development methodology, we provide a much more efficient and formal approach that those currently used for Internet routers development.

4. FUZZY INTERNET TRAFFIC CONTROLLERS

A number of research results have been reported on the application of fuzzy systems to the general area of active queue management [8,18,3] as well as traffic control with support for differentiated quality of service [23,21,1]. Recent results have also been made on the development of a fuzzy queueing theory [22] as an extension to classical queueing theory, which is the basis of many traffic processing mechanisms in the current Internet.

The aforementioned fuzzy traffic controllers manage packet queues and are targeted at core routers. These controllers show some similarities to classic real time regulators, such as PD controllers. Basically, the inputs to the fuzzy system are two: packet queue current size and packet queue variation. The fuzzy inference system must produce as output the routing decision to apply to the last packet received at the router. Some authors have applied fuzzy-PD techniques to Internet traffic control [5].



X Axis ei Y Axis ei_1 Z Axis drop Samples 60

Figure 5: Control surface of BestEffortAQM.

We have considered proposals by independent authors but preferred to develop new systems that take advantage of the prototyping platform, methodology and tool chain outlined in previous sections. We have developed Mamdani fuzzy controllers considering simplicity and reduced number of rules as main design constrains. The motivation being that because of the complexity of Internet traffic control it is not enough to design and adjust systems so that that they can exhibit near-optimal performance within a small collection of simulated scenarios. Instead, major objectives are interpretability, flexibility and adaptability, which require testing on a broad range of complex simulated and real scenarios. We also note that simplicity in terms of the number of rules will ease wide-scale adoption of fuzzy controllers as experimental Internet traffic controlling systems.

As an example we outline one of the systems developed, named BestEffortAQM (Active Queue Management for best effort service class), which implements a traffic controller for congestion control on routers which only best-effort class of service. The rule base is presented in table 1 whereas the resulting control surface is depicted in figure 5.

Two inputs and one output are defined, as described above. Input e_i is the deviation between the number of currently queued packets and a desired value reference, while input e_{i-1} is the deviation at the last time interval. 7 linguistic terms are defined for both inputs, ranging NVB, NB, NS, Z, PS, PB, PVB, for increasing differences. The output of the system, p_i, is defined as a probability value for discarding the next packet to be routed. 7 linguistic terms are defined for increasing levels of probability ranging Z, T, VS, S, B, VB, H.

p_i		e_{i-1}						
		NVB	***NB***	***NS***	***Z***	***PS***	***PB***	***PVB***
e_i	***NVB***	H	H	H	H	H	H	H
	NB	B	B	B	VB	VB	H	H
	NS	T	VS	S	S	B	VB	VB
	Z	Z	Z	Z	T	VS	S	B
	PS	Z	Z	Z	Z	T	T	VS
	PB	Z	Z	Z	Z	Z	Z	T
	PVB	Z	Z	Z	Z	Z	Z	Z

Table 1: AQMBestEffort Rule Base.

A number of fuzzy controllers for Internet traffic have been developed (implementation results are given in the next section). The performance of this new fuzzy systems as well as other independent proposals and traditional traffic controllers has been compared in terms of stability, convergence speed, fairness and other criteria in a set of network scenarios.

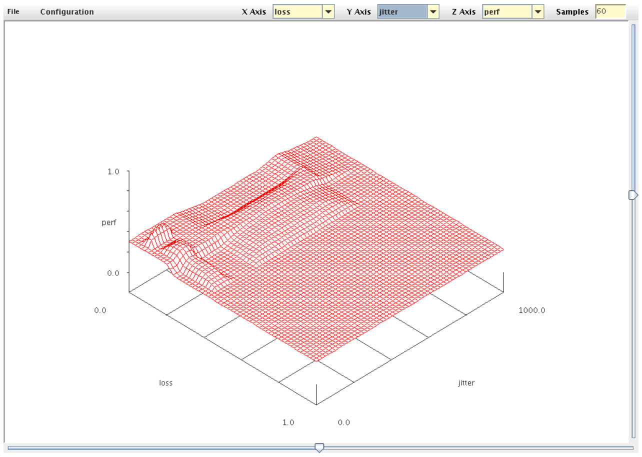
The following systems are considered in this paper and their implementations will be detailed in the next section:

- RxBufferSize, a controller for dynamic reception buffer adjustment targeted at real-time traffic.
- BestEffortAQM, a queue controller for networks with no class of service supports.
- DSSelect, a classifier for class of service enabled networks.
- AQMDSAF, a queue controller for assured forwarding class of service traffic flows.
- AQMDSBE, a queue controller for best-effort class of service traffic flows.
- RTPerf, for fuzzy inference of current network status suitability for real-time traffic.

Through the development it has been possible to systematically refine our fuzzy controllers to simulate and outperform other controllers in a wide set of conditions and with an overall higher adaptability.

As an additional example, one of the control surfaces of RTPerf is shown in figure 6. RTPerf has a 4-dimensional input space (inputs are current estimates of packet one-waydelay, loss, round-trip-time and jitter). Figure 6 shows the control surface for packet loss and delay jitter. Output is a coefficient of suitability of current network conditions to real-time traffic. The output of RTPerf is used as input to complementary fuzzy traffic controllers within the DiffServ differentiated services scheme [1,23].

The fuzzy systems described fit (depending on its operational layer) into two subsystems commonly found on Internet routers:



X Axis loss

Y Axis jitter

Z Axis perf

Samples 60

Figure 6: Control surface of RTPerf.

- For those systems that perform direct control of packet forwarding queues, a fuzzy processing module must be integrated for each output queue. Nevertheless, most current routers implement the so called virtual output queueing (VOQ) technique [10,7], so the fuzzy controller would be integrated into the circuitry of each input port . Those are systems such as RxBufferSize, BestEffortAQM, AQMDSAF and AQMDSBE.
- For higher layers processing within the DiffServ [1] architecture, fuzzy systems must be implemented as additional engines in the central network processing unit. Those are systems such as DSSelect and RTPerf.

5. FPGA BASED IMPLEMENTATION RESULTS

We detail the results of the microelectronic implementation of a set of controller prototypes on FPGAs. The focus is on the implementation results for FIM modules as they are the key component with higher operational requirements. Prototypes have been implemented on a Xilinx Spartan III FPGA, xc3s1500-fg456-5 device (1.5 millions of equivalent gates) included in the development board employed, an AvNet ADS-XLX-SP3-EVL1500.

xfvhdl was used to generate VHDL descriptions from XFL specifications as described in section 3. xfvhdl provides several FIM implementation options. In particular, we set ROM based storage for the rule base and membership functions.

A performance evaluation of the FIM architecture in terms of inference speed, area and power consumption was conducted. Synthesis and implementation was performed using Xilinx ISE 6.xx, tools xst G.28 (for synthesis) and par G.28 (for place and routing) both using default configuration.

As for precision, a number of configurations were evaluated, in particular results are presented for the following two:

- 5 bits for encoding inputs and outputs, 6 bits for encoding membership functions (556 henceforward).
- 8 bits for encoding inputs and outputs, 8 bits for encoding membership functions (888 henceforward).

The 888 configuration was found to introduce no significant errors in tests done through simulation and emulation. Better precision configurations were evaluated and found to provide no better performance than the 888 configuration. Prototypes were validated to match the correct behavior of software implementations by using simulated and emulated scenarios, as explained in section 3.

Table 2 shows a set of properties for each of the considered controllers. These properties be seen as an indicator of complexity. Properties are: inputs, linguistic terms (for inputs and output) and number of rules in canonical form. We note though that the shape membership functions introduces an additional complexity factor.

System	***Inputs***	***Linguistic terms***	***Rules (canonical form)***
RxBufferSize	2	5,5,5	25
BestEffortAQM	2	7,7,7	37
DSSelect	2	5,5,2	14
AQMDSAF	2	3,3,4	7
AQMDSBE	2	3,3,4	9
RTPerf	4	5,5,5,5,5	27

Table 2: Characteristics of fuzzy controllers.

The columns in table 3 and table 4 show a number of metrics of area occupation for the fuzzy systems being considered. LUT refers to look-up-tables in the Spartan III family of FPGAs from Xilinx. These are post-implementation results as reported by the par tool.

System	***Equivalent gates***	***Slices***	***Slice flip flops***	***LUTs***
RxBufferSize	6627	135	108	219
BestEffortAQM	6823	150	110	254
DSSelect	6564	130	105	211
AQMDSAF	6585	130	105	211
AQMDSBE	6492	128	104	212
RTPerf	7641	208	102	367

Table 3: Area results (556 precision).

System	***Equivalent gates***	***Slices***	***Slice flip flops***	***LUTs***
RxBufferSize	9363	387	138	424
BestEffortAQM	10625	391	140	471
DSSelect	9784	367	137	453
AQMDSAF	9103	375	137	382
AQMDSBE	9150	370	136	390
RTPerf	13790	767	142	844

Table 4: Area results (888 precision).

Table 5 shows for both precision configurations the number of input/output blocks (IOB) occupied as well as the achievable inference rate in MFIPS.

Power consumption is always below 335 mW.

System	***556 precision***		***888 precision***	
	IOBs	***MFIPS***	***IOBs***	***MFIPS***
RxBufferSize	19	164.2	28	125.3
BestEffortAQM	19	102	28	91.9
DSSelect	19	156	28	130.3
AQMDSAF	19	164.4	28	133.4
AQMDSBE	19	162.1	28	130.7
RTPerf	29	122.1	44	113.9

Table 5: I/O Blocks and inference rate.

Overall conclusions regarding the described microelectronic implementation results can be summarized as follows:

The amount of complexity introduced into a router system is negligible as compared to the complexity increment that is taking place at present and will happen in foreseeable high performance routers [10,13]. On the other hand, our solution provides a development methodology and a tool chain that fulfill an important gap in current custom, unscalable and inefficient design schemes [13]. In fact, an FPGA approach to the implementation of router components is in line with the current trend towards FPGA based development router design of major vendors [7,10, 13].

Power consumption of prototype controllers, between 13 mW and 335 mW, is negligible as compared to mean power consumption of current routing equipment (three orders of magnitude below the overall consumption of a current high performance router). Mean power consumption is 4.7 kW for a typical Cisco 12816 equipment. Power consumption of Juniper routers belonging to M and T series ranges from 0.58 kW up to 6.5 kW [12].

As for inference speed, prototypes implemented on a Xilinx Spartan III FPGA could achieve inference speeds around 100 MFIPS. Routers from the Cisco 12000 series process up to 25 millions of packets per second per input port. Thus, even our prototype implementation on middle cost FPGAs can provide the required inference speed in current high performance routers.

Thus, the described prototypes satisfy operational requirements of current high performance routing equipment in terms of correct behavior, complexity, precision, inference rate and power consumption.

6. CONCLUSIONS AND FUTURE WORK

Research results lead to the conclusion that fuzzy systems can help solve current problems in Internet traffic control. However, efficient hardware implementations are required to fulfill common operational requirements. We have presented a prototyping platform as well as a methodology and companion tool chain for the development and microelectronic implementation of fuzzy controllers for Internet traffic.

A set of traffic controllers have been implemented on FPGAs as prototypes which have been validated to perform correctly and better than current traffic control schemes. Implementation results were shown to satisfy operational requirements of current and future high performance routing hardware in terms of both inference speed and resource consumption. In addition, the prototypes have been designed for easy integration with routing architectures currently deployed in the Internet.

Note that introducing the fuzzy set theory formalism and a high level formal language for traffic controllers specification opens a research area on the improvement of traffic controllers by means of systematic adjustment techniques developed for fuzzy systems. These include advanced fuzzy operators and automatic learning and adjustment techniques such as those based on genetic algorithms and neural networks.

The open prototyping platform together with the methodology and tool chain presented pave the way for further development of efficient intelligent traffic controllers but also foster the development of fuzzy systems for a number of areas where intelligent analysis systems are sought, such as packet and flow identification, classification and filtering, among many others.

7. REFERENCES

[1] S. Blake, D.L. Black, M.A. Carlson, E. Davies, Z. Wang,W. Weiss, “An Architecture for Differentiated Services,” RFC 2475, Internet Engineering Task Force, Network Working Group, December 1998. Category: Informational. http://www.ietf.org/html.charters/diffserv-charter.html.

[2] A. Cabrera, S. Sánchez-Solano, P. Brox, A. Barriga, R. Senhadji. “Hardware/Software Codesign of Configurable Fuzzy Control Systems”. *Applied Soft Computing*, Vol. 4, n. 3, pp. 271-285, December 2004.

[3] H.C. Cho, M.S. Fadali, H. Lee, “Dynamic Queue Scheduling using Fuzzy Systems for Internet Routers,” IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) 2005, Reno, USA, May 2005.

[4] Cisco Systems, Inc. “Cisco Series 12000 router performance evaluation. 2005. http://www.cisco.com/warp/public/765/tools/quickreference/routerperformance.pdf.

*[5] G. Di Fatta, F. Hoffmann, G.L. Re, A. Urso. “*A Genetic Algorithm for the Design of a Fuzzy Controller for Active Queue Management,” *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 33 (3), pp. 313–334, August 2003.*

[6] M. Dolenc, T. Markovic. *PCI IP Core Specification*. Rev. 1.2, OpenCores Organization, July 2004. http://www.opencores.org/pdownloads.cgi/list/pci?no_loop=yes.

[7] W.J. Goralski, *Juniper and Cisco Routing. Policy and Protocols for Multivendor IP Networks.* Wiley Publishing Inc., Indianapolis, Indiana, 2002. ISBN: 0-471-21592-9.

[8] Y. HadJadj-Aoul, A. Nafaa, D. Negru, A. Mehaoua. “FAFC: Fast Adaptive Fuzzy AQM Controller for TCP/IP Networks,” IEEE Global Telecommunications Conference. Dallas, Texas, USA, November 2004.

[9] R. Herveille et al. *WISHBONE System-on-Chip (SoC) Interconnection Arhitecture for Portable IP Cores*, Revision B.3, OpenCores Organization, September 2002. http://www.opencores.org/projects.cgi/web/wishbone/wishbone.

[10] M. Hidell, P. Sjödin, O. Hagsand, "Control and Forwarding Plane Interaction in Distributed Routers,". Technical Report TRITA-S3-LCN-0501, Laboratory for Communication Networks, Department of Signals, Sensors, and Systems. KTH Royal Institute of Technology, Stockholm, Sweden, March 2005. http://web.it.kth.se/~mahidell/.

[11] Information Sciences Institute, University of Southern California, *The Network Simulator – ns-2,* November 2005. http://www.isi.edu/nsnam/ns/.

[12] Juniper Networks Inc., "Juniper Networks M-series Multiservice Edge Routing Portfolio," November 2005. http://www.juniper.net/products/mseries/.

[13] D. Lampret *et al. OpenRISC 1000 Architecture Manual*. OpenCores Organization, June 2005. http://www.opencores.org/projects.cgi/web/or1k.

[14] N. McKeown. "Growth in Router Capacity," IPAM Workshop on Large-Scale Communication Networks. Lake Arrowhead, CA, USA, October 2003. http://tiny-tera.stanford.edu/˜nickm/talks/index.html.

[15] F.J. Moreno-Velo, I. Baturone, S. Sánchez-Solano, A. Barriga. "Rapid Design of Fuzzy Systems With Xfuzzy," FUZZ-IEEE03. The 12th IEEE International Conference on Fuzzy Systems, vol. 1, pp. 342-347. May 2003. http://www.imse.cnm.es/Xfuzzy/xfpapers.html.

[16] F.J. Moreno-Velo, S. Sánchez-Solano, I. Baturone A. Barriga , D.R. López, "XFL3: a New Fuzzy System Specification Language," 5th WSES/IEEE Multiconference on Circuits, Systems, Communications and Computers (CSCC'01), pp. 361-366. Rethymon, July 2001. http://www.imse.cnm.es/Xfuzzy/xfpapers.html.

[17] OpenCores Organization. *OpenCores.Org: Free Open Source IP Cores and Chip Design*. http://www.opencores.org, November 2005.

[18] R. Resende, N. Nassif, A. Silva, M. Siqueira, A.E. da Silva, M. Lima-Marques, "Quality of Service control in IP networks using Fuzzy Logic for Policy Condition Evaluation," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) 2005, Reno, USA, May 2005.

[19] R. Usselmann *et al., WB/OPB & OPB/WB Interface Wrapper: Overview*, September 2004. http://www.opencores.org/projects.cgi/web/opb_wb_wrapper.

[20] C. Wang , R. Xiao, K. Sohraby, B. Li, S. Li. "Active queue management based on fuzzy logic," *Journal of Information and Computational Science*, February 2005. http://comp.uark.edu/~cgwang/html/publication.html.

[21] M. H. Yaghmaee. "Design and Performance Evaluation of a Fuzzy Based Traffic Conditioner for Differentiated Services," Computer Networks, 47 (6), pp. 847-869, January 2005.

[22] R. Zhang, Y.A. Phillis, V.S. Kouigoglou. *Fuzzy Control of Queuing Systems.* Springer-Verlag, Surrey, UK, January 2004. ISBN: 1-85233-824-5

[23] R. Zhang, Y.A. Phillis, J.Ma. "A Fuzzy Approach to the Balance of Drop and Delay Priorities in Differentiated Services Networks," *IEEE Transactions on Fuzzy Systems*, 11 (6), pp. 840-846, December 2003. http://www.sics.se/˜runtong/.