Intelligent Scheduling of Aggregate Traffic in Internet Routers by Means of Fuzzy Systems

Federico M. Pouzols\textsuperscript{1} Diego R Lopez\textsuperscript{2}
\textsuperscript{1}Microelectronics Institute of Seville - CNM-CSIC
Avda. Reina Mercedes s/n. Edif. CICA.
E-41012 Sevilla, Spain
fedemp@imse.cnm.es diego.lopez@rediris.es

\textsuperscript{2}RedIRIS, Spanish NREN
Planta 2. E-28020 Madrid, Spain
barriga@imse.cnm.es santiago@imse.cnm.es

Abstract

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. This paper presents novel mechanisms for intelligent traffic scheduling in Internet routers by means of fuzzy logic based systems. A systematic design methodology, interpretability principles, evaluation over a broad range of network scenarios as well as practical implementation constraints have been considered. A comparative evaluation of results obtained by means of our fuzzy controllers as compared to that of traditional approaches is outlined\textsuperscript{1}.

Keywords: Fuzzy control, Aggregate traffic, Congestion control, Quality of service, Queue scheduling, Active queue management.

1 Introduction

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 \cite{16}.

Quality of service requirements as well as traffic patterns of emergent services and applications are difficult to characterize and demand deep advances in current flow and congestion control schemes. Because of the nature of these problems (complexity, no feasible analytic solution as well as incomplete and inaccurate information) the employment of intelligent systems based on fuzzy logic and other soft computing techniques is an appealing alternative.

Currently deployed schemes for traffic regulation in Internet (as well as proposed alternatives) fit into one of the two following approaches \textsuperscript{2}:

- Distributed control, with functionality distributed among the end nodes in the network and implemented by means of end-to-end transport protocols. Transmitter and receiver end nodes of packet flows cooperate so as to perform congestion control and fair distribution of network resources.

- Queue schedulers in intermediate nodes (routers). These mechanisms can discriminate packet flows and enforce resource distribution and reservation.

Thus, regulation of packet flows from sender to receivers is performed on both an end-to-end and a per-hop basis and can involve all the

---

\textsuperscript{1} This work has been partially supported by projects TEC2005-04359/MIC from the spanish Ministry of Education and Science as well as TIC2006-635 from the Andalusian regional Government.

\textsuperscript{2}
network nodes in the end-to-end path. This leads to a system with complex interactions which comprises multiple feedback loops, as shown in figure 1.

Both aforementioned approaches can be redefined in terms of fuzzy systems, which does not only provide a deeply backgrounded engineering approach but also a modelling and analysis framework for Internet traffic (which the current Internet research community lacks [16]). This paper focuses on traffic regulation of aggregate traffic in routers. We present novel mechanisms for intelligent (in the sense of flexibility and adaptability) traffic scheduling in routers by means of fuzzy logic based systems.

2 Intelligent Scheduling of Aggregate Traffic

The dominant queue scheduling scheme in the current Internet is the passive FIFO queue without classes of service (known as drop-tail), that discards packets when the storage space is full. Active schemes (known as AQM - Active Queue Management-) are however being developed and promoted [2, 18] since AQM mechanisms are required to provide quality of service, differentiate services or penalize misbehaving flows, among other demanded functionalities.

Current Internet routers at core networks process aggregate traffic [12] which typically comprises millions of packets per second as well as millions of active end points and simultaneous flows established by services and applications with an increasing diversity of traffic patterns. Analytically modelling these aggregates at core routers is a challenging task. Furthermore, those scheduling architectures that require state information for all active flows are eventually not deployable at global scale because of its complexity. It is thus assumed that Internet routers schedulers must cope with a high degree of uncertainty.

Although a number of AQM schemes have been proposed [14], properties of aggregate traffic [12] (such as self-similarity and burstiness at multiple time scales) make it difficult to stabilize packet queues. There are a great deal of challenges in tuning AQM schemes in real environments and no generally accepted solution has been found.

To cope with the aforementioned problems, we introduce the notion of FAQM (fuzzy active queue management). By means of FAQM we aim at defining aggregate traffic schedulers that can perform in a flexible and adaptive manner.

Figure 2 shows a scheme of a fuzzy scheduler in an output queue of an Internet router. A Mamdani fuzzy inference system regulates a variable number of packet queues. In the most basic scheme, inputs (linguistic variables of the rule base antecedents) are queue sizes as well as its variation whereas the output variable is defined as a probability value or reference to determine which packet should be sent next. This scheme eases the integration of intelligent traffic analysis systems as inputs to the scheduler.

3 Approach and Related Work

A general theory of fuzzy systems for queueing control [17, 15] has been developed. A number of proposals of fuzzy systems for specific areas of network traffic control have been made. Among these, [18] describes fuzzy systems for balancing priorities when considering multiple classes of service within the DiffServ architec-
ture [2]. Results in optimizing flow control in asynchronous transfer mode (ATM) based B-ISDN networks by means of Takagi-Sugeno fuzzy controllers are reported in [3]. In [6] limited simulation results for balancing service rates among classified queues have been reported. Genetic algorithms have been successfully employed for optimizing queue controllers [8]. A number of additional works that deal with the application of fuzzy systems to the area of active queue management as well as traffic control can be found in the literature (see for example [7]). Our work differs from previous results regarding the following points:

- Simulation is performed through the ns-2 [11], a de facto standard within the Internet research community. Realistic state-of-the-art models of Internet aggregate traffic are considered.
- We use a design methodology [3] and tool chain [13] for the whole development process that cover from initial high-level description to implementation as software and hardware components. The methodology and tool chain are overviewed in the next section.
- Practical implementation constraints are considered, (current protocols, implementations and technological constraints). In particular, efficient hardware implementations that can achieve the high inference rates required by current and future high performance Internet links [9].

Figure 3: Fuzzy Systems Design Flow and Tool Chain

4 Development Methodology and Tool Chain

Resulting from more than a decade 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. Leveraging on the Xfuzzy [13] CAD suite of tools and a methodology [3] for the development of fuzzy controllers, we have defined a methodology and tool chain tailored for the development of fuzzy Internet traffic schedulers.

The design flow and tool chain employed to develop fuzzy inference modules is depicted in figure 3. The whole development process is covered, from initial description to final implementation whether as software or hardware. The first development stage (description) is performed using a high level fuzzy systems specification language, XFL [13], which can be automatically turned into C and VHDL code among other implementation options.

The development stages after specification have been tailored for Internet traffic control as follows. For network simulation, we have used ns-2 [11]. 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
5 Fuzzy Internet Traffic Scheduling of Aggregate Traffic

Fuzzy Internet traffic schedulers can be developed as replacements for traditional traffic schedulers proposed for Internet routers. These employ three non-exclusive mechanisms in order to regulate traffic [10, 2]:

- Basic AQM, which attempt to prevent congestion by discarding packets when queues grow.
- Explicit congestion notification (ECN), whereby the scheduler sends control (notification) packets back to senders and/or intermediate routers in case of congestion.
- Admission control, i.e., filtering of packets that match an admission criteria (such as flow rate and source/destination subnetwork).

Any of these mechanisms imply that some packets are selected to trigger a certain action. Those packets which are selected are said to be marked by the scheduler. Depending on protocols and architectures, marking can correlate to one or more of the following actions: modify packet headers, discard packet (do not forward to next node in the network), send ECN notifications and activate filtering rules.

Among traffic controllers proposed for Internet routers, the most widely accepted is RED [10], an AQM scheduler which discards packets so as to enforce end-to-end traffic regulation. Though a number of variants and specializations of RED have been defined, they are generally based on the definition of a queue length threshold and a discard probability value that is proportional to queue length. According to figure [2] we propose a fuzzy scheduler that marks packets the same way as RED schedulers, i.e., discarding them. We distinguish buffer size and queue length values following recent developments on the subject [1].

This kind of fuzzy schedulers 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 forwarding decision to apply to the next packet in the queue.

Mamdani fuzzy controllers were developed considering simplicity and low number of rules as main design constraints. Because of the complexity of Internet traffic control it is not enough to design and adjust systems so 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 compatibility with deployed infrastructures and simplicity in terms of the number of rules ease the adoption of fuzzy systems as experimental Internet traffic schedulers.

6 Fuzzy Scheduler of Best-Effort Aggregate Traffic

This section describes the design of the FAQMBestEffort fuzzy system, which has been developed as a scheduler for best-effort traffic according to the AQM paradigm. FAQMBestEffort implements a traffic controller for congestion control on routers with no support for classes of service.

Two inputs and one output are defined. An scheme of membership functions for both inputs is shown in figures [4] and [5] which depict the fuzzy variable types $T_{e_i}$ and $T_{e_{i-1}}$, respectively. Input $e_i$ is the deviation between the number of currently queued packets and
a desired value reference, while input $e_{t-1}$ is the deviation at the last time interval.

The output of the system, $p_t$, is defined as a probability value for marking the next packet to be forwarded. In this case, as in AQM schemes currently most accepted within the Internet research community, marking is defined as dropping the packet. An scheme of membership functions for the fuzzy type $T_p$ is shown in figure 6.

The rule base is presented in table 1 whereas the resulting control surface is depicted in figure 7. 7 linguistic terms are defined for both inputs, ranging from NVB to PVB for increasing differences. As for the output variable, 7 linguistic terms are defined for increasing levels of probability ranging from Z to H.

Table 1: FAQMBestEffort Rule Base.

<table>
<thead>
<tr>
<th>$e_{t-1}$</th>
<th>NVB</th>
<th>NB</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PB</th>
<th>PVB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVB</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>NB</td>
<td>T</td>
<td>Z</td>
<td>H</td>
<td>Z</td>
<td>T</td>
<td>Z</td>
<td>T</td>
</tr>
<tr>
<td>NS</td>
<td>Z</td>
<td>T</td>
<td>Z</td>
<td>Z</td>
<td>T</td>
<td>Z</td>
<td>T</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>PS</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>T</td>
<td>Z</td>
<td>T</td>
</tr>
<tr>
<td>PB</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>T</td>
</tr>
<tr>
<td>PVB</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>T</td>
</tr>
</tbody>
</table>

In order to simplify definition and easing the employment of efficient implementation techniques, only triangular and trapezoidal membership functions are used. Fuzzy inference follows the Mamdani model, and the center of mean defuzzification method is employed to compute crisp output values. Membership functions were adjusted following typical performance values considered in recent Internet measurement studies [16].

7 Comparative Performance Analysis

A performance evaluation study was conducted on traditional and fuzzy traffic schedulers for routers so as to compare both approaches. What follows is a summary of results from the FAQMBestEffort scheduler as compared to the results from a RED system in a simulation scenario which resembles a typical network configuration where traffic schedulers can have a direct impact on overall network performance for end users.

The following network scenario is consid-
Red: we analyze the behavior of the RED scheduler at a router that resembles a campus access point (were current network configurations usually lead to a higher degree of performance degradation as seen by end users [4]). The full network scenario considers higher scale regional, national and international links, comprising end-to-end paths up to 15 hops long. Traffic is generated from a number of network end points following state-of-the-art models for Internet traffic [16] as a way to generate as realistic as possible aggregate traffic with a high degree of statistical multiplexing at the router under analysis.

A performance comparison of both RED and FAQMBestEffort schedulers is outlined in terms of a number of metrics: queue size, throughput (goodput), packet delay distribution. We will not detail additional properties and operational constraints (such as router load and inference rate) which were also analyzed within a complete feasibility study.

Figures 8 and 9 show the evolution of queue length over a typical period of 10 seconds for the RED and the FAQMBestEffort schedulers, respectively. Queue length oscillations are significantly lower for the FAQMBestEffort scheduler which also manages to keep a higher mean queue length. Overall statistical properties of the oscillations of both cases are summarized in table 2, which shows mean queue length, standard deviation, maximum peak value, minimum peak value, and 5% and 95% percentiles.

The higher stability of FAQMBestEffort can also be seen in figures 10 and 11 which show the evolution of the output (marking value) for both schedulers.

Application level throughput (goodput) resulting from both schedulers is compared in figure 12. As expected, the higher stability and mean queue length in the FAQMBestEffort case lead to higher throughput.

The results summarized above imply that end users will experience a higher mean delay in the FAQMBestEffort. The delay increase is nevertheless negligible for current network technologies and indeed generally lower than

<table>
<thead>
<tr>
<th>Scheduler</th>
<th>Mean</th>
<th>Stdv</th>
<th>Max</th>
<th>Min</th>
<th>5 pc</th>
<th>95 pc</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED</td>
<td>24.6</td>
<td>9.6</td>
<td>50</td>
<td>0</td>
<td>9.1</td>
<td>41.1</td>
</tr>
<tr>
<td>FAQM</td>
<td>33.8</td>
<td>4.6</td>
<td>46</td>
<td>20</td>
<td>26.1</td>
<td>41.1</td>
</tr>
</tbody>
</table>

Figure 8: Queue length (RED scheduler)

Figure 9: Queue length (FAQMBestEffort scheduler)

Figure 10: RED Scheduler Output

Figure 11: FAQMBestEffort Scheduler Output
5% of the overall end-to-end delay in our simulation scenario. On the other hand, FAQMBestEffort improves peak delay values as compared to those of RED schedulers by approximately 50%, which implies a significant improvement in end-to-end jitter. Thus, FAQMBestEffort performance is better for best effort traffic while the benefits it introduces for real-time traffic clearly outperforms the hardly noticeable mean delay increase.

We note that, although developed for bulk transfer traffic with no time constraints, the higher degree of robustness and responsiveness to packet bursts shown by FAQMBestEffort leads to an improvement in end-to-end performance as experienced by time constrained services and applications. FAQMBestEffort is thus a practical compromise solution for currently deployed routers.

We also note that light variations of the fuzzy scheduler (mostly through membership functions shifts) can provide results suited for specialized traffic. It is thus easy to develop schedulers specialized for particular traffic patterns. These schedulers can be combined in an intelligent manner within class of service enabled infrastructures.

As a general conclusion, results from FAQMBestEffort show a higher robustness in the presence of self-similar bursty traffic and outperforms results from RED for both bulk transfer and real-time traffic, showing better performance in terms of queue length and stability, link utilization, as well as impact on end-to-end delay and jitter.

8 Conclusions and Future Work

The FAQM scheme for scheduling aggregate traffic in Internet routers has been outlined. Within the FAQM scheme, results from FAQMBestEffort show a higher robustness than traditional traffic schedulers in the presence of self-similar bursty traffic and outperforms RED results for both bulk transfer and real-time traffic, showing better performance in terms of queue length, stability, utilization, delay and jitter, among other parameters.

We are currently working on the application of fuzzy systems learning and adjustment techniques to gain further insight on aggregate traffic scheduling. As future research we also plan to extend the fuzzy scheduler described with the addition of fuzzy systems for class of service discrimination as well as traffic analysis systems as additional inputs.

References


