Fuzzy Logic Congestion Control in TCP/IP Differentiated Services Networks for Quality of Service Provisioning

C. Chrysostomou, A. Pitsillides, G. Hadjipollas

Centre for Telecommunications and Information Engineering, Monash University, Melbourne, Australia.
e-mail: ahmet@titania.ctie.monash.edu.au

C. Chrysostomou, A. Pitsillides, G. Hadjipollas

Department of Computer Science*, University of Cyprus, 75 Kallipoleos Street, P.O. Box 20537, 1678 Nicosia, Cyprus.
e-mail: {cchrys, andreas.pitsillides, hpollas}@ucy.ac.cy

A. Sekercioglu

Centre for Telecommunications and Information Engineering, Monash University, Melbourne, Australia.
e-mail: ahmet@titania.ctie.monash.edu.au

M. Polycarpou

Department of Electrical and Computer Engineering, University of Cyprus, 75 Kallipoleos Street, P.O. Box 20537, 1678 Nicosia, Cyprus.
e-mail: mpolycar@ucy.ac.cy

Abstract

This paper presents a new active queue management scheme that provides congestion control in TCP/IP networks using a fuzzy logic control approach. The proposed scheme is implemented within the differentiated services (Diff-Serv) framework, providing quality of service (QoS). It is based on the Fuzzy Explicit Marking (FEM) controller proposed recently to provide congestion control in TCP/IP best-effort networks. The provision of QoS in a Diff-Serv environment requires an adequate differentiation between assured and best-effort classes of service in the presence of congestion, giving priority-preference to assured-tagged traffic. For this purpose, a two-class FEM controller, called FEM In/Out is presented. The proposed fuzzy logic approach for congestion control allows the use of linguistic knowledge to capture the dynamics of nonlinear probability marking functions, uses multiple inputs to capture the (dynamic) state of the network more accurately, and can offer effective implementation. A simulation study over a wide range of traffic conditions shows that the FEM In/Out controller outperforms the Random Early Detection (RED) implementation for Diff-Serv in terms of link utilization, packet losses, and queue fluctuations and delays. Also, the proposed scheme can offer adequate differentiation among assured and best-effort traffic.

1. Introduction

The rapid growth of the Internet and increased demand to use the Internet for time-sensitive applications necessitate the design and utilization of new network architectures to include more effective congestion control algorithms in addition to the standard TCP based congestion control. As a result, the Differentiated Services (Diff-Serv) architecture was proposed [1] to deliver (aggregated) quality of service (QoS) in IP networks. Recently, active queue management (AQM) mechanisms [2] have been proposed to provide high network utilization with low loss and delay (e.g. random early detection – RED [3]). RIO [4], a RED implementation within the framework of the Diff-Serv architecture, was proposed to preferentially drop packets.

The AQM approach can be contrasted with the “Tail Drop” (TD) queue management approach, employed by common Internet routers, where the discard policy of arriving packets is based on the overflow of the output port buffer. Contrary to TD, AQM mechanisms [2] start dropping packets earlier in order to be able to notify traffic sources about the incipient stages of congestion. AQM allows the router to separate policies of dropping packets from the policies for indicating congestion. The use of Explicit Congestion Notification (ECN) [5] was proposed in order to provide TCP an alternative to packet drops as a mechanism for detecting incipient congestion in the network. An AQM-enabled gateway can mark a packet either by dropping it or by setting a bit in the packet’s header if the transport protocol is capable of reacting to ECN. The use of ECN for notification of congestion to the end-nodes generally prevents unnecessary packet drops.

In this paper, we present a fuzzy logic based approach for delivering an improved and more predictable congestion control implementation within the Diff-Serv architecture. The application of fuzzy control techniques to the problem of congestion control in networks is suitable due to the difficulties in obtaining a precise mathematical model using conventional analytical methods, while some intuitive understanding of
congestion control is available.

In this paper we use fuzzy logic techniques to develop a new AQM scheme, implemented within the Diff-Serv framework - using a two-class FEM controller (FEM In/Out – FIO) - to provide congestion control. The Fuzzy Explicit Marking (FEM) controller was proposed recently [6] to provide congestion control in TCP/IP best-effort networks. The proposed fuzzy control system is designed to regulate the queues of IP routers at a predefined level, by achieving a specified target queue length (TQL), in order to maintain both high utilization and low mean delay. A fuzzy inference engine (FIE) is designed to operate on router buffer queues, and uses linguistic rules to mark packets in TCP/IP networks. In a Diff-Serv framework a two-class FEM controller is designed to operate on the core routers' buffer queues, called FEM In/Out (FIO). Two identical FEM controllers are used, one for each differentiated class of service (that is, assured and best-effort), and two different TQLs are introduced, one for each FEM controller. The proposed fuzzy logic strategy is shown via simulations to be robust with respect to traffic modeling uncertainties and system nonlinearities, yet provide tight control. As a result, it can effectively regulate the queues of the bottleneck links, while achieving high utilization, low loss and delay. It also achieves an adequate differentiation between the two traffic classes of service in the presence of congestion, by preferentially marking the lowest-priority packets, while controlling the queue at the predefined levels, and providing QoS.

The paper is organized as follows. Section 2 discusses key aspects of the Diff-Serv architecture, congestion control and AQM. In Section 3 we briefly review some of the properties of fuzzy logic control and present our proposed fuzzy logic based congestion control scheme, namely FIO. Then Section 4 presents simulation examples and discusses the performance of FIO. Finally in Section 5 we present our conclusions.

2. Differentiated Services – congestion control – AQM mechanisms

Since Integrated Services failed to be adopted for widespread use, the IETF (Internet Engineering Task Force) proposed a more evolutionary approach that did not require significant changes to the Internet infrastructure and provided differentiation of services (Diff-Serv) [1].

The Diff-Serv working group has defined two broad aggregate behavior groups: the Expedited Forwarding (EF) Per-Hop Behaviour (PHB) and the Assured Forwarding (AF) PHB. The EF-PHB can be used to build a low loss, low latency, low jitter, assured bandwidth end-to-end service, thus indirectly providing some QoS. The AF-PHB group provides delivery of IP packets in four independently forwarded AF classes. Within each AF class two or three drop preference levels are used to differentiate flows. The idea behind AF is to preferentially drop best-effort packets and non-contract conforming packets when there is congestion. By limiting the amount of AF traffic in the network and by managing the best-effort traffic appropriately, routers can ensure low loss behavior to packets marked with the EF PHB.

AQM mechanisms have recently been proposed [2], with the aim to provide high link utilization with low loss rate and queuing delay, while responding quickly to load changes. Several schemes have been proposed to provide congestion control in TCP/IP networks. RED [3], which was the first AQM algorithm proposed, simply sets some minimum and maximum marking thresholds in the router queues. In case the average queue size exceeds the minimum threshold, RED starts randomly marking packets based on a probability depending on the average queue length, whereas if it exceeds the maximum threshold every packet is dropped.

Diff-Serv will try to provide some QoS using a Diff-Serv aware congestion control algorithm. The most popular algorithms used for Diff-Serv implementation, to preferentially drop non-contract conforming against conforming packets, are based on RED [3]. The RED implementation for Diff-Serv, called RED In/Out (RIO) [4], defines different thresholds for each class of service. RIO uses the same mechanism as in RED, but is configured with two different sets of parameters, one for “In” packets, and one for “Out” packets. The discrimination against “Out” packets is created by carefully choosing the parameters of minimum and maximum thresholds, and maximum marking probability. Best-effort (“Out”) packets have the lowest minimum and maximum thresholds, and therefore they are marked earlier than packets of Assured class (“In”). They are also marked with a higher probability by setting the maximum marking probability higher than the one for packets of Assured class.

The properties of RED and its variants have been extensively studied in the past few years, and many issues of concern have been arisen (see for example [7, 8]).

3. Fuzzy logic control – FIO implementation for Diff-Serv

3.1. Fuzzy logic

Fuzzy logic is one of the tools of what is commonly known as Computational Intelligence (CI). CI is an area of fundamental and applied research involving numerical information processing. We are beginning to see a lot of interest using such techniques not only from the academic research community (e.g. [9]), but also from industry (e.g. [10]). Fuzzy Logic Control (FLC) may be viewed as a way of designing feedback controllers in situations where rigorous control theoretic approaches cannot be used due to difficulties in obtaining a formal
analytical model, while at the same time some intuitive understanding of the process is available. The control algorithm is encapsulated as a set of linguistic rules. FLC has been applied successfully (e.g. [11]) for controlling systems in which analytical models are not easily obtainable or the model itself, if available, is too complex and possibly highly nonlinear.

In recent years, a number of research papers using fuzzy logic investigating solutions to congestion control issues in networking, especially to ATM networks, have been published (e.g. [12]). A survey is given in [9]. Furthermore, fuzzy logic is recently applied [6] to TCP/IP best-effort networks providing congestion control.

3.2. FIO implementation for Diff-Serv

Our design of a fuzzy control system is based on a fuzzy logic controlled AQM scheme - namely Fuzzy Explicit Marking (FEM) [6]. FEM controller [6] was proposed to provide congestion control in TCP/IP best-effort networks. In this paper our design is implemented within the Diff-Serv framework. For this purpose, a two-class FEM controller – FEM In/Out (FIO) - is designed and used for Diff-Serv control.

The system model of FEM [6] is shown in Figure 1, where all quantities are considered at the discrete instant \( kT \), with \( T \) the sampling period; \( e(kT) = q_{\text{des}} - q \) is the error on the controlled variable queue length, \( q \), at each sampling period; \( e(kT - T) \) is the error of queue length with a delay \( T \) (at the previous sampling period); \( p(kT) \) is the mark probability; and \( SG_i \) and \( SG_o \) are scaling gains.

The proposed fuzzy control system is designed to regulate the queues of IP routers by achieving a specified desired TQL, \( q_{\text{des}} \), in order to maintain both high utilization and low mean delay. A fuzzy inference engine (FIE) is designed to operate on router buffer queues, and uses linguistic rules to mark packets in TCP/IP networks. As shown in Figure 1, the FIE dynamically calculates the mark probability behavior based on two network-queue state inputs: the error on the queue length (i.e., the difference between the desired (TQL) and the current instantaneous queue length) for two consecutive sample periods (which can be interpreted as a prediction horizon). The FEM controller is implemented with marking capabilities, so that FEM-like routers have the option of either dropping a packet or setting its ECN bit in the packet header, instead of relying solely on packet drops (for the rest of the paper, by marking a packet it is meant setting its ECN bit). The decision of marking a packet is based on the mark probability, which is dynamically calculated by the FIE.

The scaling gains, \( SG_i \) and \( SG_o \), shown in Figure 1, are defined as the maximum values of the universe of discourse of the FIE input and output variables, respectively. In order to achieve a normalized range of the FIE input variables from -1 to 1, the input scaling gain \( SG_i \) is set to be equal to \(-1/(q_{\text{des}} - \text{QueueBufferSize})\), if the instantaneous queue length is greater than the TQL; otherwise \( SG_i \) is equal to 1. The output scaling gain \( SG_o \) is determined so that the range of outputs that is possible is the maximum, but also ensuring that the input to the plant will not saturate around the maximum. \( SG_o \) is set to a value indicating the maximum mark probability that can also be adjusted in response of changes of the queue length.

The FIE uses linguistic rules to calculate the mark probability based on the input from the queues (see Table 1). Usually multi-input FIEs can offer better ability to linguistically describe the system dynamics. The dynamic way of calculating the mark probability by the FIE comes from the fact that according to the error of queue length for two consecutive sample periods, a different set of fuzzy rules, and so inference apply. Based on these rules and inferences, the mark probability is calculated more dynamically than the classical RED approach.

This point can be illustrated by observing the visualization of the decision surface of the FIE used in the FEM scheme (see Figure 2). An inspection of this decision surface and the linguistic rules shown in Table 1 provides hints on the operation of FEM. The mark probability behaviour under the region of equilibrium (i.e., where the error on the queue length is close to zero) is smoothly calculated. On the other hand, the rules are aggressive about increasing the probability of packet marking sharply in the region beyond the equilibrium point. These rules reflect the particular views and experiences of the designer, and are easy to relate to human reasoning processes and gathered experiences.

Usually, to define the linguistic values of a fuzzy variable, Gaussian, triangular or trapezoidal shaped membership functions are used. Since triangular and

---

Table 1. Linguistic rules – Rule base

<table>
<thead>
<tr>
<th>( p(kT) )</th>
<th>( Q_{\text{error}} (kT - T) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVB</td>
<td>Z</td>
</tr>
<tr>
<td>NB</td>
<td>B</td>
</tr>
<tr>
<td>NS</td>
<td>T</td>
</tr>
<tr>
<td>PS</td>
<td>Z</td>
</tr>
<tr>
<td>PB</td>
<td>Z</td>
</tr>
<tr>
<td>PVB</td>
<td>Z</td>
</tr>
</tbody>
</table>

\( Q_{\text{error}} (kT - T) \)

<table>
<thead>
<tr>
<th></th>
<th>NVB</th>
<th>H</th>
<th>H</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>Z</td>
<td>PS</td>
<td>PB</td>
<td>PVB</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>Z</td>
<td>PS</td>
<td>PB</td>
<td>PVB</td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td>T</td>
<td>VS</td>
<td>S</td>
<td>S</td>
<td>B</td>
<td>PB</td>
<td>PVB</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>T</td>
<td></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>Z</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 content notations: negative/positive very big (NVB/PVB), negative/positive big (NB/PB), negative/positive small (NS/PS), zero (Z), huge (H), very big (VB), big (B), small (S), very small (VS), tiny (T).
trapezoidal shaped functions offer more computational simplicity, they have been selected for the rule base (see Figure 3). Then, the rule base is fine tuned by observing the progress of simulation, such as packet marking and delay occurrences, and throughput curves. The tuning can be done with different objectives in mind. For example, any gain in throughput must be traded off by a possible increase in the delay experienced at the terminal queues. Alternatively, an adaptive fuzzy logic control method [13] can be used, which is based on tuning the parameters of the fuzzy logic controller on line, using measurements from the system. The tuning objective can be based on a desired optimization criterion, for example, a trade-off between maximization of throughput with minimization of end-to-end delay experienced by the users. This is part of our future work.

In a Diff-Serv framework, a two-class FEM controller is designed to operate on the core routers’ buffer queue, called FEM In/Out (FIO), where “In” and “Out” terms are used to distinguish packets that are classified into different classes of service, such as assured and best-effort classes. Both assured and best-effort packets share a FIO queue. FIO comprises of two identical FEM controllers, one for each class of service (that is, assured and best-effort), and we introduce two different TQLs, on the total queue length, one for each FEM controller. The TQL for best-effort is lower than the one for assured traffic. That is, best-effort packets are more likely to be marked than the assured ones. The idea behind this is to regulate the queue, if possible, to the lower TQL, in order to get a mark probability for the assured traffic closed to zero. In the presence of large amount of assured traffic, compared with the one of best-effort traffic, the queue can be regulated at the higher TQL, where the mark probability for best-effort traffic would be closed to one. Therefore, we can accomplish a bounded delay, by regulating the queue between the two TQLs, depending on the dynamic network traffic conditions.

FIO can achieve an adequate differentiation between the two classes of service in the presence of congestion, by preferentially marking the lowest-priority best-effort packets, and giving priority/preference to assured-tagged traffic, while controlling the queue at the predefined levels, and providing QoS.

The design of the proposed fuzzy logic based congestion control system aim to generally provide better congestion control and better utilization of the network, with lower losses and delays than the classical RED approach [4], especially by introducing additional input variables and on-line (dynamic) adaptivity of the rule base (self-tuned).

4. Performance evaluation – simulation results

In this section we evaluate the performance and robustness of the proposed fuzzy logic based scheme, namely FIO, in a wide range of environments, and compare with other published results by taking RIO [4] in the case of a TCP/IP Diff-Serv network, using a recent version of NS-2 [14] simulator. We have conducted a series of simulations in order to evaluate the performance of both FIO and RIO schemes, and examine their capabilities to provide QoS. Two classes of service are set: Assured traffic class, which has the
highest priority, and best-effort traffic class, which has the lowest priority in a buffer queue.

All results are summarized in Table 2, where the performance-QoS metrics are the bottleneck link utilization, the loss rate and the mean queuing delay with its standard deviation.

The sampling period for FIO AQM is fixed to 0.006 sec. The TQL for best-effort traffic is set to 100 packets, whereas the TQL for assured traffic is set to 200 packets, for a buffer size of 500 packets. For RIO, the minimum and maximum thresholds, for best-effort traffic, are set to 50 and 150 packets, respectively. The equivalent values for assured traffic are 100 and 300 packets, respectively. The maximum mark probability for best-effort traffic is set to 0.1, whereas the one for assured traffic is set to 0.02, for both FIO and RIO.

4.1. Scenarios I

The network topology used for Scenarios I is shown in Figure 4. We use TCP/Newreno with an advertised window of 240 packets. The size of each packet is 1000 bytes. The buffer size of all queues is 500 packets. We use AQM in the queues of the bottleneck link between router-A and router-B. All other links have a simple Tail Drop queue. All sources (N flows) are greedy sustained FTP applications. The link capacities and propagation delays are set as follows: \((C_1, d_1) = (100\text{Mbps}, 5\text{ms})\), \((C_2, d_2) = (15\text{Mbps}, 120\text{ms})\), and \((C_3, d_3) = (200\text{Mbps}, 5\text{ms})\), while \(N = 100\). The simulation time is 100 sec.

Scenario I-1 considers a limited number of flows tagged as assured class traffic; 2 out of 100 flows are considered belonging to assured class, whereas the rest, 98 flows, are tagged as best-effort. Figure 5 shows the queues of both FIO and RIO, where we can observe that FIO regulates its queue to the lower TQL (100 packets), whereas RIO exhibits very large queue fluctuations that results in degraded utilization, losses and high variance of queuing delay (see Table 2). Furthermore, FIO achieves an adequate differentiation between the two traffic classes, in contrast with RIO that cannot provide sufficient link utilization for assured class traffic.

Scenario I-2 increases the number of flows tagged as assured traffic class to 10. FIO accomplishes a bounded queuing delay, between the two TQLs, that results in high link utilization and minimal losses (see Figure 6 and Table 2). On the other hand, RIO slowly regulates its queue, after a significant transient period with large overshoots that results in degraded utilization, losses and high variance of queuing delay (see Table 2). Furthermore, FIO achieves a much higher differentiation between the two traffic classes, as compared with RIO, thus can provide adequate QoS.

Scenario I-3 examines the behavior of the AQM schemes under dynamic traffic changes. We use the previous experiment, and provide some time-varying dynamics by stopping the assured-tagged flows at time \(t = 40\text{ sec}\), and resuming transmission at time \(t = 70\text{ sec}\). The results (see Figure 7) show that FIO is very robust against the dynamic traffic changes and keeps very good
response. Between $t = 40 – 70$ sec, where only best-effort-tagged flows are active, FIO successfully manages to regulate the queue length at the TQL for best-effort, whereas RIO fails to do so.

Scenario I-4 increases the number of flows tagged as assured traffic to 90. In the presence of large amount of assured traffic, compared with the best-effort traffic, FIO regulates its queue at the higher TQL (see Figure 8). RIO, on the other hand, exhibits large queue fluctuations that result in lower utilization and higher losses than FIO has.

Scenario I-5 uses the previous experiment, and examines the effect of the RTT by having heterogeneous propagation delays of the links between the sources and router-A (we separate the 100 flows into groups of 10, and for each group - that consists of 9 assured-tagged flows and 1 best-effort-tagged flow – its propagation delay is increased by 5 msec, starting from 5 msec up to 50 msec). The propagation delay of the bottleneck link has also changed to 60 msec. The results (see Figure 9) show the superior steady performance of FIO with stable queue length dynamics that result in a high link utilization with minimal losses, while RIO exhibits large queue fluctuations, worst than the previous experiment, that result in a significant amount of losses and high variance of queuing delay.

4.2. Scenario II

Scenario II uses the network topology shown in Figure 10 with the introduction of TCP/Web-like traffic too. In this scenario, we use TCP/SACK with an advertised window of 240 packets. The size of each packet is 1514 bytes. The buffer size of all queues is 500 packets. We use AQM in the queues of the bottleneck link between router-B and router-C. The link capacities and propagation delays are set as follows: $(C_1, d_1) = (500Mbps, 30ms)$, $(C_2, d_2) = (100Mbps, 10ms)$, $(C_3, d_3) = (10Mbps, 30ms)$, $(C_4, d_4) = (100Mbps, 10ms)$, and $(C_5, d_5) = (500Mbps, 1ms)$, while $N = 100$. The simulation time is 1500 sec.

In this experiment (Scenario II) web traffic is considered belonging to assured traffic class, whereas ftp traffic is considered belonging to best-effort traffic class. The number of flows tagged as assured traffic class is set to 10, while the rest (90 flows) are tagged as best-effort traffic class.

In case of web traffic the standard deviation of the queuing delay is mainly determined by the burstiness of the arriving traffic. Due to the high traffic variability, there exists a large possibility for high queuing delays. Even with these circumstances, FIO manages to maintain better the queue around the higher TQL (200 packets), in contrast with RIO (see Figure 11). Furthermore, FIO appears to control better the flow rate across the network and provide a more stable behaviour. From the results (see Table 2) it can be seen that FIO can provide the necessary congestion control and differentiation and ensure acceptable QoS in a Diff-Serv network. It achieves a better discrimination between the two traffic classes than RIO does, whilst maintaining high utilization and minimal losses.
5. Conclusions

We have presented a new AQM scheme implemented in TCP/IP networks - within the differentiated services framework - using fuzzy logic techniques to provide effective congestion control by achieving high utilization, low losses and delays. The proposed scheme, which we refer to as Fuzzy Explicit Marking In/Out (FIO) – a two-class FEM controller -, is contrasted with the classical RED approach through a wide range of scenarios.

The proposed fuzzy logic approach for congestion control is implemented with marking capabilities (either dropping a packet or setting its ECN bit). In this paper the design of the fuzzy knowledge base is kept simple, using a linguistic interpretation of the system behavior. We have successfully used the reported strength of fuzzy logic and have addressed limitations of existing RED approach implemented in a Diff-Serv framework. This is clearly shown from the simulative evaluation. The FIO controller is shown to exhibit many desirable properties, like robustness and fast system response, and behave better than RED variant (RIO) in terms of queue fluctuations and delays, packet losses, and link utilization, with capabilities of adapting to highly variability and uncertainty in network. The FIO controller also achieves an adequate differentiation between the two classes of service (assured and best-effort) by preferentially marking the lowest-priority packets, while controlling the queue at the predefined levels.

We believe that future work can include the design of a fuzzy model reference learning controller, which can tune the parameters of the fuzzy logic controller on line, using measurements from the system, to obtain even better behavior. Furthermore, it is worth investigating the implementation of a multi-class FEM controller in a mobile IP-based radio access network (RAN), and its integration with proposed resource management frameworks in such environments, like the resource management in Diff-Serv (RMD) framework [15].

From the results presented we are optimistic that the Fuzzy Control methodology can offer significant improvements on controlling congestion in TCP/IP differentiated services networks.

6. References