Using Fuzzy Logic Control to address challenges in AQM Congestion Control in TCP/IP Networks

C. Chrysostomou and A. Pitsillides

Department of Computer Science,
University of Cyprus,
75 Kallipoleos Street, P.O. Box 20537,
1678 Nicosia, Cyprus.
Tel.: +357 22892700
Fax: +357 22892701
e-mail: {cchrys, andreas.pitsillides}@ucy.ac.cy

Abstract: Network management and control is a complex problem that requires robust, possibly intelligent, control methodologies to obtain satisfactory performance. Active Queue Management (AQM) mechanisms have been introduced to assist the TCP congestion control. We discuss the modeling and control approach followed by a number of representative AQM schemes, and address possible limitations to meet the diverse needs of today’s Internet. Fuzzy Logic Control is adopted due to its reported strength in controlling non-linear systems using linguistic information. Emphasis is given towards the ability of effectively controlling the congestion in TCP/IP networks.

I. INTRODUCTION

The rapid growth of the Internet and increased demand to use the Internet for time-sensitive applications with differing Quality of Service requirements (e.g. VoIP, video streaming, Peer-to-Peer, interactive games) necessitate the design and utilization of new network architectures, including more effective congestion control algorithms, to supplement the standard TCP based congestion control. It should also be noted that, even for the present Internet architecture, network congestion control remains a critical and high priority issue, and is unlikely to disappear in the near future. Furthermore, if we consider the current utilization trends, congestion in the Internet may become unmanageable unless effective, robust, and efficient methods for congestion control are developed. The development of such effective congestion control protocols would benefit from the cooperation between networking and control researchers.

Recently, many active queue management (AQM) schemes have been proposed to provide high network utilization with low loss and delay by regulating queues at the bottleneck links in TCP/IP networks, including random early detection (RED) [11], adaptive RED (A-RED) [12], proportional-integral (PI) controller [13], random exponential marking (REM) [45] and fuzzy explicit marking (FEM) [4-7].

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 [43] start dropping packets earlier in order to notify traffic sources about the incipient stages of congestion (TCP interprets dropped packets as congestion). AQM allows the router to separate policies of dropping packets from the policies for indicating congestion. The use of Explicit Congestion Notification (ECN) [44] was also proposed in order to provide TCP an alternative to packet drops as a mechanism for detecting incipient congestion in the network. In this case packets are not dropped, rather a bit is set on their header indicating congestion, and returned via the destination to the source. The ECN scheme requires both end-to-end and network support. 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, an overview of a number of representative AQM mechanisms is presented. In particular, we choose:

- The RED scheme, as being the traditional AQM mechanism in the networking research community, which can be seen as a linear heuristic-based AQM technique.
- The PI controller, which is a linear control theory based technique.
- The FEM controller, which is a nonlinear fuzzy logic based technique.

We describe the control approach that each of the representative AQM schemes follows. Furthermore, we discuss a number of models, already proposed in literature, that use a control-theoretic approach for design and analysis of a TCP/AQM system. These models are used to analyse the AQM system and provide guidelines/recommendations for correct configuration of AQM parameters that lead to stable and robust operation, as well as to illustrate, in some cases, the difficulty in setting AQM parameters to stabilize TCP. From the discussion, we address possible limitations of these models to meet the diverse needs of today’s Internet due to the dynamic, time-varying nature of TCP/IP networks. Such limitations are:

- The dependency of AQM control parameters on dynamic network parameters, like the number of flows and the round trip time.
- The accuracy of the proposed TCP/AQM models, as they ignore the slow start phase of TCP and/or timeout events that are prominent conditions in today’s Internet with the existence of short-lived TCP/Web flows.
- The linearity of the control functions of proposed AQM mechanisms that cannot capture effectively the nonlinearities of the TCP network.

On the other hand, the concept of Fuzzy Logic Control is adopted as a candidate of AQM mechanism due to its reported strength in controlling non-linear systems using linguistic information. Fuzzy Logic Control is an appealing alternative to conventional control methods when systems follow some general operating characteristics that can be linguistically described, and a detailed process understanding is unknown or traditional systems models become overly complex [10]. The capability to qualitatively capture the attributes of a control system based on observable phenomena is a main feature of fuzzy logic control and has been demonstrated in various research literature and commercial products. The main idea is that if the fuzzy logic control is designed with a good (intuitive) understanding of the system to be controlled, the limitations due to the complexity system’s parameters introduce on a mathematical model can be avoided. A common approach is to either ignore such complex parameters in the mathematical model, or to simplify the model to such an extent (in order to obtain some stability results), which render the designed controllers and their derived stability bounds overly conservative.

In this paper, we demonstrate through extensive simulations over a wide range of network conditions that the fuzzy logic based AQM technique better handles the nonlinearities of the TCP network, and thus provides an effective control to congestion.

The paper is organized as follows: Section II discusses the modelling and control approach followed by the chosen AQM schemes and addresses possible limitations. Section III presents simulation examples evaluating the performance of the AQM schemes, and Section IV presents important remarks of this study. Finally, Section V presents the main conclusions of this extensive study of AQM, and ends with some thoughts about possible future work.

II. AQM Mechanisms

AQM mechanisms 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. The main AQM performance characteristics include [13]:
• Efficient queue utilization: the queue should avoid overflow that results in lost packets and undesired retransmissions or emptiness that results in link underutilization.
• Queuing Delay: It is desirable to keep small both the queuing delay and its variations.
• Robustness: AQM scheme needs to maintain robust behaviour in spite of varying network conditions, such as variations in the number of TCP sessions, and variations in the propagation delay and link capacity.

A discussion of a number of representative AQM schemes is followed, in terms of modelling and control approach, and possible limitations are addressed.

A. Linear Heuristic based Technique - RED

Random Early Detection (RED) [11], 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 linear heuristic-based control law (see Equation 1 and Figure 1), with a drop/marking probability depending on the average queue length, whereas if it exceeds the maximum threshold every packet is dropped.

\[
P = \begin{cases} 
0, & \text{if } q_{\text{avg}} < q_{\text{min}} \\
\frac{q_{\text{avg}} - q_{\text{min}}}{q_{\text{max}} - q_{\text{min}}} p_{\text{max}}, & \text{if } q_{\text{min}} \leq q_{\text{avg}} \leq q_{\text{max}} \\
1, & \text{if } q_{\text{avg}} > q_{\text{max}}
\end{cases}
\]

(1)

where \(q_{\text{min}}\) and \(q_{\text{max}}\) are the lower and upper thresholds, and \(p_{\text{max}}\) is the maximum drop/mark probability. The average queue length, \(q_{\text{avg}}\), is updated at every packet arrival according to the exponentially weighted moving average (EWMA) method, as shown in Equation 2:

\[
q_{\text{avg}}^{\text{new}} = (1-w) \times q_{\text{avg}}^{\text{old}} + w \times q_{\text{inst}}
\]

(2)

where \(q_{\text{inst}}\) is the instantaneous queue length, and \(w\) is the averaging weight, \(0 \leq w \leq 1\).

RED is designed to avoid “global synchronization”, a case where all active sources reduce their sending rates at the same time that results in a fluctuation of link utilization, by introducing randomized packet dropping/marking [11].

The properties of RED and its variants have been extensively studied in the past few years. It is becoming clear that for successful implementation of RED based AQM (or its variants) in TCP/IP networks, there are still a number of unresolved issues. These include:

• Problems with performance of RED under different scenarios of operation and loading conditions. For example, the influence of: the moving average of the queue occupancy on the responsiveness of control; the loss function on the congestion control feedback mechanism; and the buffer size to the reaction time of the congestion controller is documented, see for example [18]. Note that most of the existing studies are obtained with simulations suggesting modifications to the algorithm. Numerous RED variants (e.g. [19, 20, 21]) have been proposed, perhaps motivated by the difficulty in understanding the dynamics of RED completely. In [22] a comparative evaluation of RED with standard parameter settings, RED with optimal parameter settings [23], and Gentle RED [24] found: no performance improvements with RED compared to Tail Drop that may justify deployment of RED in current backbone by ISPs; performance of RED with standard parameters setting exhibits higher dependency on the load situation than other mechanisms, indicating that fine tuning of the RED parameters is not sufficient to cope with undesired RED behavior. It is worth noting that Gentle RED addressed many of the RED deficiencies, but many serious deficiencies still remain.
• Tuning of RED parameters has been an inexact science for sometime now, so much so that some researchers have advocated against using RED, in part because of this tuning difficulty [25, 31]. Recently, some effort was undertaken to evaluate the performance of RED analytically. In [25] the dependence of RED performance with respect to bursty traffic is studied. An approach to investigate RED in the presence of feedback traffic is presented in [26] where a source consists of a 3-state Markov model. In [27] authors investigated issue of recommendations of RED parameters and derived a set of recommendations of RED parameters for configuration of the RED queue size estimator: the frequency of queue sampling and the average weight. In [28] a more formal, control theoretic stand-point is adopted to also investigate issue of recommending settings for the RED parameters. In [23] quantitative models on how to set RED parameters with TCP traffic are derived. The correct tuning of RED implies a “global” parameterisation that is very difficult, if not impossible to achieve as it is shown in [29].

In [19, 28, 31] it is stated that one of RED’s main weaknesses is that the average queue size varies with the level of congestion and with parameter settings. As a result, the mean queuing delay from RED is sensitive to the traffic load, as well as to the RED parameters. Consequently the throughput is also sensitive to the traffic load and to RED parameters. The authors of RED themselves have identified later on in [12] that “RED does not perform well when the average queue becomes larger than max threshold, resulting in significantly decreased throughput and increased dropping rates. Avoiding this regime would require constant tuning of the RED parameters”.

Adaptive-RED (A-RED) [12], proposed by the author of RED [11], attempts to solve the problem for the need of tuning RED parameters by modifying a similar proposal [20]. In particular, A-RED adjusts the value of the maximum mark probability to keep the average queue size within a target range half way between the minimum and maximum thresholds. Thus, A-RED maintains a desired average TQL twice the minimum threshold (if the maximum threshold is kept three times the minimum threshold). The adjustment of the maximum mark probability is based on an additive fixed increase step when the average queue length exceeds the desired average queue, and on a multiplicative fixed decrease step when the average queue length goes below the desired average value, following a linear AIMD approach. Furthermore, A-RED also specifies a procedure for automatically setting the RED parameter of average queue weight as a function of the link capacity, following the approach in [23].

However, in [30] it is demonstrated that the fixed increase step size for the adjustment of the maximum drop/mark probability affects the performance of A-RED algorithm with loose convergence to the target average queue, as it is hard to quickly respond to traffic changes. The study in [30] shows that for different traffic conditions, different values of the increase step size would be preferred.

A-1. Control-Theoretic Design and Analysis of TCP/RED

Recently, some effort was undertaken to evaluate the performance of RED-like AQM mechanisms analytically using control theory. For this purpose, the TCP/AQM flow dynamics are modelled and analysed in terms of feedback control theory. The design of such models is used to analyse the AQM system and provide guidelines/recommendations for correct configuration of AQM parameters that lead to stable and robust operation, as well as to illustrate, in some cases, the difficulty of setting AQM parameters to stabilize TCP for the control approaches under study.

As shown in Figure 2, the TCP congestion control dynamics with an AQM scheme can be modelled as a feedback control system. It consists of (i) a plant - the controlled process – with TCP sources, receivers, and routers, (ii) an AQM controller – the controlling element – which generates as a control signal the packet drop/mark probability, $p$, (iii) the controlled
variable queue length, \( q \), and (iv), the feedback signal, which is a system output observed information.

In [27], RED implementation is studied and several structural problems are identified, such as large traffic oscillations. The system model introduced in [27] is a deterministic nonlinear dynamic feedback system model, consisting of a derived plant function (queue law), averaging function, and RED control function. The exact form of the plant function depends on system parameters such as the number \( N \) of connections, the nature of the connections, and round-trip delays. Potential problems of instability of the feedback system are identified, and therefore, a set of recommendations for configuration of the RED control law (frequency of queue sampling and the average queue weight) are derived.

The authors in [28] have developed a system of nonlinear differential equations for TCP/AQM dynamics using fluid-flow analysis that ignores the TCP slow start phase. Using the derived model, the role played by the RED configuration parameters on the behavior of the algorithm in a network is explained. It is believed that the RED averaging mechanism is a cause of tuning problems for RED.

A combined TCP and AQM model is analyzed from a control theoretic standpoint in [34]. Linearization is used to analyze a previously developed nonlinear model of the system [28]. The forward-path transfer function of the plant depends on the number of connections, the round trip time, the link capacity, and a time delay factor. The analysis on an AQM system implementing RED is performed. Design guidelines are presented for choosing parameters that lead to stable operation of the linear feedback control system.

In [37-38] the dynamics of TCP over RED queues through linearization around equilibrium points are studied. The main conclusion of this paper is that “it is stability that largely determines the dynamics of TCP/RED”, and not only because of noise traffic or its Additive Increase/Multiplicative Decrease (AIMD) probing. A general nonlinear model of TCP/RED is developed, and linearization of the model is done around the equilibrium in order to study local stability. The linear model generalizes the model of [13]. The results taken suggest that TCP/RED becomes unstable when delay increases, or when link capacity increases. The analysis done illustrates the difficulty of setting RED parameters to stabilize TCP: they can be tuned to improve stability, but only at the cost of large queues, even when they are dynamically adjusted.

A nonlinear modeling and analysis framework is followed in [40]. A deterministic nonlinear dynamical model for a simplified TCP network with RED control is used. This work goes beyond a simple linear stability analysis and studies regions where nonlinear instabilities occur due to the nonlinearity in the TCP throughput characteristic as a function of drop probability at the gateway. The model is shown to exhibit a rich variety of bifurcation behavior, leading to irregular operation. As system parameters are varied, the system dynamics are shown to transit between a stable fixed point and oscillatory and/or chaotic behavior (instability). In particular, the effects of various system and control parameters (such as, the averaging weight \( w \), the lower queue threshold \( q_{\text{min}} \), the number of connections, and the round trip delay) on average queue behavior are studied, as each of these parameters is varied while the others are fixed. The results show that as the averaging weight is increased a chaos-type phenomenon is given. Also, the system becomes less stable as \( q_{\text{min}} \) is increased. On the other hand, the system stabilizes as the number of connections increases, though at the expense of increased delay. Furthermore, larger round
trip propagation delays cause instability. The oscillatory behavior appearing in the system is due to the inherent nonlinearity of the interaction between RED mechanism and TCP. A different approach is followed in [41], where the above mentioned problems of a TCP/RED system are due mainly to the linearity of the drop/mark probability function of the RED algorithm, and not a matter of appropriate tuning of RED parameters. Several papers have introduced models (as explained above) that aim to give appropriate configuration of RED parameters (such as averaging weight, minimum and maximum thresholds, maximum drop/mark probability) that can lead to stable and robust operation. However, as discussed earlier, it is a difficult task to find an appropriate configuration set that can be applied to broad network conditions. A major weakness of these models is that the selection of appropriate parameter set is done for a specific operating point for which various system’s parameters are assumed to be known (such as the number of flows, and the round trip time). As stated in [41], even if the assumptions regarding the input parameters fit the specific scenario, the applicability of the RED algorithm would be restricted to a small range of the assumed values only. Therefore, the configured parameter set and stability conditions introduced by the proposed models lack of applicability to all possible real scenarios with varying dynamics of network conditions. As observed in [41], during high load condition a disproportionately higher drop/mark probability is required than in a low load condition, in order to keep the queue length in the same range, a requirement met only by a nonlinear drop/mark function. Such nonlinear function fulfills the requirement of the queue length to remain between $q_{\text{min}}$ and $q_{\text{max}}$ for a much broader load conditions than the original linear RED function. Therefore, it is concluded that the linear drop/mark probability function of RED itself is not robust enough for the highly bursty network traffic. The motivation should be to find a proper nonlinear function, rather than to find appropriate tuned RED parameters for a specific operating point for the original linear RED function.

### B. Linear Control Theory Based Technique – PI

Hollot et al. uses classical control system techniques to develop controllers well suited for TCP/AQM system [13]. Three key network parameters - the number of TCP sessions, link capacity and round-trip time – are related to the underlying feedback control system. It is determined that the queue averaging in RED algorithm is not beneficial. It also recommends an alternative AQM scheme, a proportional-integral (PI) control, on managing queue utilization and delay. The key feature is that PI control allows one to explicitly set the network queuing delay by introducing a desired queue length. Using a linearized TCP/AQM dynamics, the PI controller has been proposed not only to improve responsiveness of the TCP/AQM dynamics but also to stabilize the router queue length around the desired value. The latter can be achieved by means of integral control, while the former can be achieved by means of proportional control using the instantaneous queue length instead of using the average queue length.

The authors use a simplified version of a previously developed dynamic TCP model [28] (that uses fluid-flow and stochastic differential equation analysis), which ignores both slow start phase and timeout mechanism. This model is described by the coupled, nonlinear differential equations shown in Equation 3 [13].

\[
\begin{align*}
\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t) - R(t)}{R(t) - R(t)} p(t - R(t)) \\
\dot{q}(t) &= \begin{cases} 
-C + \frac{N(t)}{R(t)} W(t), & q > 0 \\
\max \left\{0, -C + \frac{N(t)}{R(t)} W(t)\right\}, & q = 0
\end{cases}
\end{align*}
\]

where $\dot{x}$ denotes the time-derivative, $W$ is the average TCP window size, $q$ is the average queue length, $N(t)$ is the number of TCP sessions, $R(t)$ is the round-trip time, which is equal to $\frac{q(t)}{C} + T_p$, $C$ is the link capacity, and $T_p$ is the propagation delay. The first differential
equation in (3) describes the TCP window control dynamic, and the second equation in (3) models the bottleneck queue length.

Using this simplified model, linearization about an operating point is done to approximate these dynamics. To linearize (3) the authors have assumed that the number of TCP sessions is constant, that is, \( N(t) = N \). Given the vector of network parameters \((N, C, T_p)\) the set of feasible operating points \((W_o, q_o, p_o)\) is defined. The linearization about the operating point results in Equation 4 [13]. As indicated in [13] the linearization of the queue dynamic does not yield a pure integrator, but produces a leaky integrator with time constant \( R_o \). This is explained by noting that the queue’s arrival rate \( \frac{NW}{R_o} \) is a function of the round-trip time which, in turn, is a function of the queue length due to the queuing delay \( \frac{q}{C} \).

\[
\delta W(t) = -\frac{N}{R_o^2 C} (\delta W(t) + \delta W(t - R_o)) - \frac{1}{R_o C} (\delta q(t) + \delta q(t - R_o)) - \frac{R C^2}{2N^2} \delta p(t - R_o)
\]

\[
\delta q(t) = \frac{N}{R_o} \delta W(t) - \frac{1}{R_o} \delta q(t)
\]

where \( \delta W = W - W_o \), \( \delta q = q - q_o \), \( \delta p = p - p_o \) represent the perturbed variables about the operating point.

A PI control law implementation with the TCP/AQM dynamic is shown in Figure 3 [13]. Implementing the PI controller in AQM-enabled routers, results in a difference equation, at time \( t = kT \), where \( T = \frac{1}{f_s} \) is the sampling period, as shown in Equation 5.

\[
p(kT) = a \delta q(kT) - b \delta q((k-1)T) + p((k-1)T)
\]

Design rules are given in [13] for a PI controller for the linear control system to identify PI parameters. It is found that the PI controller stabilizes the feedback control system for all \( N \geq N^- \) and all \( R \leq R^+ \), where \( N^- \) is considered to be a lower bound on the number of flows, and \( R^+ \) the upper bound on round-trip time. That is, by stabilizing against the largest expected \( R_o \) and \( C \), and against the smallest expected \( N \) can lead to a robust AQM design.

In [42] an analysis for the lower bound of response time for PI, among others, is presented. As stated, since stability and response time are often in conflict with each other in system performance, PI scheme tries to find a tradeoff between them. If network parameters (especially the number of flows and the round-trip time) are known a priori, PI can adjust its control constant parameters to obtain the best response properly while guaranteeing stability. However, this is not the case, since in a dynamic network it is difficult to obtain these parameters precisely. The analysis done shows that the response time of PI is dependent on their control constant parameters: buffer size, desired queue length, and desired stable packet drop/mark probability, \( p_o \), which is an increasing function of the number of TCP flows and a decreasing function of round-trip time and link capacity. Under heavy congestion, PI suffers from a long response time (which is the opposite of what one wants during a congestion). In addition, with small buffer size, the responsiveness of PI will become worse as well.
C. Nonlinear Fuzzy Logic Based Technique - FEM

C-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. While these techniques are not a panacea (and it’s important to view them as supplementing proven traditional techniques), there is a lot of interest not only from the academic research community (e.g. [1]), but also from industry (e.g. [2]).

Fuzzy Logic Control (FLC) [10] denotes the field in which fuzzy set theory [15] and fuzzy inference are used to derive control laws. A fuzzy set is defined by a membership function that can be any real number in the interval [0, 1], expressing the grade of membership for which an element belongs to that fuzzy set. The concept of fuzzy sets enables the use of fuzzy inference, which in turn uses the knowledge of an expert in a field of application to construct a set of “IF-THEN” rules. Fuzzy logic becomes especially useful in capturing human expert or operator’s qualitative control experience into the control algorithm, using linguistic rules.

The idea of FLC was initially introduced by Zadeh [16], and first applied by Mamdani [17] in an attempt to control systems that are difficult to control mathematically. FLC may be viewed as a way of designing feedback controllers in situations where rigorous control theoretic approaches are too difficult and time consuming to use, 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, leading to algorithms describing what action should be taken based on system behaviour observations. FLC has been applied successfully (e.g. [8-9]) for controlling systems in which analytical models are not easily obtainable or the model itself, if available, is too complex and possibly highly nonlinear.

Therefore, FLC concentrates on attaining an intuitive understanding of the way to control the process, incorporating human reasoning in the control algorithm, in contrast with conventional control approaches that concentrate on constructing a controller with the aid of an analysed system model that in many cases can be uncertain, nonlinear, and subject to noises. The capability to qualitatively capture the attributes of a control system based on observable phenomena is a main feature of FLC and has been demonstrated in various research literature and commercial products. Thus, if the fuzzy logic control is designed with a good (intuitive) understanding of the system to be controlled, the limitations due to the complexity system’s parameters introduce on a mathematical model can be avoided. A common approach is to either ignore such complex parameters in the mathematical model, or to simplify the model to such an extent (in order to obtain some stability results), which render the designed controllers and their derived stability bounds overly conservative.

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. [3]). A survey is given in [1]. Moreover, fuzzy logic is recently applied [4-7] to TCP/IP best-effort and differentiated networks providing congestion control. From [4-7], it becomes apparent that the application of fuzzy control techniques to the problem of congestion control in networks is worthy of further investigation, due to the difficulties in obtaining a precise enough mathematical model, (amicable to analysis) using conventional analytical methods, while some intuitive understanding of congestion control is available.
C-2. Fuzzy Logic Control Design

In [4-7] the design of a fuzzy control system is based on a fuzzy logic controlled AQM scheme – namely Fuzzy Explicit Marking (FEM) - that provides congestion control in TCP/IP networks. The system model of FEM is shown in Figure 4, 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 high utilization, minimal losses and low mean delay and delay variation. 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 4, 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 has its minimum value (i.e., equals to the buffer size), then the error on the queue length (\( e(kT) = q_{\text{des}} - q \)) should have its minimum value of \( q_{\text{des}} \). On the other hand, when the instantaneous queue length has its minimum value, that is, zero, then the error on the queue length has its maximum value that equals to \( q_{\text{des}} \).

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 (e.g. 10%) that can also be adjusted (tuned on line using measurements from the system) in response of changes of the instantaneous queue length. The dynamic selection of \( SG_o \) is currently still under study. Note that adaptation of scaling factors, in general, is significant as it globally influences the control surface.

The FIE uses linguistic rules to calculate the mark probability based on the input from the queues (see Table 11). 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 more responsive than other AQM approaches [11-13]

and upper bounds of the queue length. When the instantaneous queue length has its maximum value (i.e., equals to the buffer size), then the error on the queue length (\( e(kT) = q_{\text{des}} - q \)) should have its minimum value of \( q_{\text{des}} = \text{QueueBufferSize} \). The on other hand, when the instantaneous queue length has its minimum value, that is, zero, then the error on the queue length has its maximum value that equals to \( q_{\text{des}} \).

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 (e.g. 10%) that can also be adjusted (tuned on line using measurements from the system) in response of changes of the instantaneous queue length. The dynamic selection of \( SG_o \) is currently still under study. Note that adaptation of scaling factors, in general, is significant as it globally influences the control surface.

The FIE uses linguistic rules to calculate the mark probability based on the input from the queues (see Table 11). 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 more responsive than other AQM approaches [11-13]

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

Table 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).

1 Table 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).
due to the human reasoning and the inbuilt non-linearity.

This point can be illustrated by observing the visualization of the decision surface of the FIE used in the FEM scheme (see Figure 5). An inspection of this nonlinear control 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 (and where congestion starts to set in and quick relief is required). 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, a range of shapes for membership functions can be used, such as Gaussian, triangular or trapezoidal shaped membership functions. Since triangular and trapezoidal shaped functions offer more computational simplicity, we have selected them for our rule base. Both the input and output membership functions are symmetric with respect to the origin (see Figure 6). Then, the rule base is fine tuned by an iterative process based on intuition, looking into observations of system performance 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 can be traded off by a possible increase in the delay experienced at the terminal queues. Currently, linguistic rules and linguistic values are set using expert knowledge, and tuned manually from system behavior observation. Alternatively, an adaptive fuzzy logic control method [14] can be used, which is based on tuning the parameters of the fuzzy logic controller on line, using measurements from the system. It is expected to allow optimal operation under any condition. 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.

The design of the proposed fuzzy logic based congestion control system allows the use of linguistic knowledge to capture the dynamics of nonlinear probability marking functions, and uses multiple inputs to capture the dynamic state of the network more accurately. It aims to generally provide effective congestion control, and thus better utilization of the network, with lower losses and delays than other AQM schemes [11-13].
The FEM controller is a Mamdani-based model [10] that better captures the network nonlinearities. Fuzzification and Defuzzification, two important operations of a fuzzy logic controller, involve the mapping of the fuzzy variables of interest to crisp numbers used by the control system. Fuzzification interprets a crisp/numeric value for the error on the queue length \((e(kT)\) or \(e(kT-T)\)) into a linguistic value (such as Negative Very Big) with a grade of membership. Defuzzification, on the other hand, gives a crisp/numeric value. It takes the aggregated fuzzy output of the rules and results in the control input to the plant. In this paper the min-operation for the implication of each rule is selected, and the aggregation of the consequences of fuzzy rules uses the max-operation to all the resulted output fuzzy sets. The calculated output control signal of the nonlinear fuzzy controller, shown in Equation 6, uses the center of gravity [10] or centroid of area of the aggregated fuzzy output set \(C\):

\[
p_k = \frac{\int y \mu_C(y) dy}{\int \mu_C(y) dy} \quad (6)
\]

where,

\[
\mu_C(y) = \max(\mu_0(y), \mu_1(y), \mu_2(y), ..., \mu_N(y))
\]

is the membership degree of \(y\) in the aggregated fuzzy set \(C\) (which is found using the max-operation over all \(N\) implicated output fuzzy sets), and \(N = \) number of linguistic rules.

The limits of integration correspond to the entire universe of discourse \(Y\) of output mark probability values. To reduce computations, we discretize the output universe of discourse \(Y\) to \(p\) values, \(Y = \{y_1, y_2, ..., y_p\}\), which gives the discrete fuzzy centroid (see Equation 7):

\[
p_k = \frac{\sum_{j=1}^{p} y_j \times \mu_C(y_j)}{\sum_{j=1}^{p} \mu_C(y_j)} \quad (7)
\]

III. Performance Evaluation – Demonstration of Limitations of Existing AQM Mechanisms

In this section we evaluate the performance and robustness of a number of representative AQM mechanisms. Based on the discussion presented in Section II, we choose the RED scheme, in particular the A-RED [12] as being the traditional AQM mechanism in the networking research community, which can be seen as a linear heuristic-based AQM technique, the PI controller [13], which is a linear control theory based technique, and the FEM controller [4] which is a nonlinear fuzzy logic based technique.

Some indicative simulation results are shown here. An extensive simulative evaluation can be found in [46]. The performance evaluation is done in a wide range of environments, examining the influence of both network and AQM parameters. In particular we examine [46] the effects of:

- dynamic traffic changes – time-varying dynamics
- traffic load factor
- heterogeneous propagation delays at access links
- increase/decrease of round trip propagation delays at bottleneck links
- introduction of noise-disturbance to the network (e.g. short-lived TCP connections)
- different types of data streams, like TCP/FTP and TCP/Web-like traffic, as well as Video and Voice over IP (unresponsive UDP traffic)
- use of single- and multiple-bottleneck links
- various target queue lengths in order to examine the sensitivity of the AQM algorithms

The performance metrics used to compare the AQM schemes are:

- Throughput (Goodput)/Utilization of the bottleneck links
- Loss rate
- Mean queuing delay and its standard deviation

The performance of the AQM schemes is evaluated using the network simulator NS-2 [47].

\[2\] In [46] we have also evaluated the performance of REM [45] AQM scheme. In general, the results show similar behaviour as PI has. The reader can refer to [4, 5, 6, 46] for details.
In this section we show indicative results obtained from both single- and multiple-bottleneck links scenarios. Scenarios I and II uses the single-bottleneck network topology shown in Figure 7, whereas Scenario III uses the multiple-bottleneck network topology shown in Figure 17. 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. The sampling period for FEM and PI AQM is fixed to 0.006 sec (following TQL of FEM and PI, except otherwise defined, is set to 200 packets, as is used in [13]). The TQL of FEM and PI, except otherwise defined, is set to 200 packets, as is used in [13]. For A-RED, we set the minimum threshold to 100 packets, and the maximum to 300, giving an average TQL of 200 packets, while the gentle parameter is enabled. The simulation time is 100 sec.

A. Scenarios I and II

The network topology used is shown in Figure 7. We use AQM in the queues of the single bottleneck link between router-A and router-B. All other links have a simple TD queue. All sources (N flows) are greedy sustained FTP applications, except for Scenario II, where we also introduce web-like traffic.

In Scenario I-1, we examine whether FEM AQM can regulate the queue to stabilize at arbitrary TQLs. Given that N = 60, \((C_1, d_1) = (15\text{Mbps}, 40\text{ms}), (C_2, d_2) = (15\text{Mbps}, 5\text{ms}),\) and \((C_3, d_3) = (30\text{Mbps}, 5\text{ms}),\) we choose the TQL to be equal to 50, 100, 200, and 300 packets. The results, shown in Figure 8, show the ability of FEM AQM to adequately regulate the queue length at the target values, and, consequently, controlling the queuing delay.

Scenario I-2 is based on Scenario I-1. We have conducted a comparison of FEM with the other AQM schemes, for a TQL of 200 packets (see Figure 9). FEM quickly regulates the queue to the reference value, while PI controller spends considerably long time. A-RED shows good control performance, however, after a significant transient period of about 10 sec with large overshoots.

In Scenario I-3 we increase the number of flows from 60 to 100, in order to explore the transient performance of the AQM schemes. The performance of the AQM schemes under dynamic traffic changes is also examined. We provide some time-varying dynamics by stopping half of the flows at time \(t = 40\text{ sec},\) and resuming transmission at time \(t = 70\text{ sec}.\) The results (see Figure 10) show that FEM is very robust against the dynamic traffic changes and keeps very good response by successfully maintaining the queue length at the target value. PI is not as robust, as it is slower to settle down to the reference value, resulting in large queue fluctuation. A-RED responds well, except for some larger overshoots at the time of the traffic changes.

In Scenario II-1, we investigate the performance of AQM schemes under higher link capacities and propagation delays, that is, we set \((C_1, d_1) = (100\text{Mbps}, 5\text{ms}), (C_2, d_2) = (15\text{Mbps}, 120\text{ms}),\) and \((C_3, d_3) = (200\text{Mbps}, 5\text{ms}),\) while \(N = 100.\)
We also keep the time-varying dynamics on the network, as used in Scenario I-3. We specifically examine the effect of the round-trip time (RTT) by increasing the propagation delay of the bottleneck link (i.e., 120 ms). In general, an increase of RTT degrades the performance of an AQM scheme. The results (see Figure 11) show the superior steady performance of FEM with stable queue length dynamics (it adapts quickly to longer RTT delays and provides a stable response), while PI, and A-RED exhibit large queue fluctuations that result in degraded utilization and high variance of queuing delay.

Scenario II-2 investigates the effect of the traffic load factor (N) in the last experiment, by increasing N from 100 to 200, 300, 400, and 500. The expected queuing delay experienced at router-A is 106.7 ms (15Mbps link capacity
corresponds to 1875 packets/sec; for a TQL of 200 packets the expected mean delay is 200/1875 = 0.1067). Figure 12 shows the loss rate as traffic load increases, where it can be seen that FEM has the lowest drops. FEM shows stable and low packet loss over large traffic load. A-RED has the largest drops with a large increase of packet loss with respect to higher loads. Figure 13 shows the utilization of the bottleneck link with respect to the mean queuing delay. FEM outperforms other AQM schemes on both high utilization and low mean delay, thus it exhibits a more stable, and robust behavior. The other AQM schemes show a poor performance as the number of traffic load increases, achieving much lower link utilization, and large queuing delays, far beyond the expected value. Table 2 lists the statistical results of the mean queuing delay and its standard deviation. It is clear that FEM has the lowest variance in queuing delay, resulting in a stable and robust behavior. On the other hand, the other AQM schemes exhibit very large queue fluctuations with large amplitude that inevitably deteriorates delay jitter.

In Scenario II-3, we further investigate the performance of AQM schemes by introducing additional web-like traffic that can be seen as
noise-disturbance to the network. In particular, we keep the same parameters as in Scenario II-I, without the time-varying dynamics. The number of flows is kept to 100 for FTP applications, with an additional 100 web-like traffic flows. We have conducted experiments for two specific values of the TQL (i.e., 100 and 200 packets) to examine the robustness of the AQM schemes. For both cases the results are shown in Table 3 where we obtain the mean queuing delay and its standard deviation, link utilization and loss rates. It is clear that, for both cases, FEM keeps a queuing delay close to the expected one with the lowest variance, while it exhibits the highest link utilization with the lowest drops. This is in contrast with the other AQM schemes that exhibit very large variations of the queue; consequently, this has the effect of having degraded link utilization with large number of drops.

We further conducted an experiment to investigate the influence of the network parameters (number of TCP flows and round trip time) on the behavior of the PI controller. As indicated in Section II-B, the PI parameters ($a$ and $b$, see Equation 5) are selected so as to stabilize the feedback control system for all $N \geq N^-$ and all $R \leq R^+$, where $N^-$ is considered to be a lower bound on the number of flows, and $R^+$ the upper bound on round-trip time. However, the PI parameters are fixed/static and it is difficult to get a stable operation in a broad range of dynamic varying traffic conditions. This is illustrated in Figure 14 and 15. In particular, Figure 14 shows the queue

| Table 2. Scenario II-2: Summary of the mean delay and standard deviation |
|--------------------------|-------------------------|-------------------------|
|                          | Mean-Delay (ms)          | Std-Deviation (ms)      |
| 100 Sources              | FEM 101.803 24.595       | PI 119.508 54.8057     |
|                         | ARD 106.531 72.8443     | ARED 106.976 21.3181   |
| 200 Sources              | FEM 111.527 29.395       | PI 168.225 96.2637     |
|                         | ARED 121.653 51.1104    |                     |
| 300 Sources              | FEM 117.466 30.806       | PI 183.278 99.527     |
|                         | ARED 150.439 66.1591    |                     |
| 400 Sources              | FEM 119.873 32.5612     | PI 194.903 94.0823    |
|                         | ARED 160.633 58.7155    |                     |

| Table 3. Scenario II-3: Summary of statistical results |
|--------------------------|-------------------------|-------------------------|
|                          | Mean-Delay (ms)          | Std-Deviation (ms)      | Utilization (%) | Loss rate (%) |
| TQL 100 (expected mean delay: 55.3 ms) | FEM 53.5075 14.0869 | PI 69.6015 44.9733 | 99.36 | 0.09 |
|                          | ARED 57.2572 42.6883   | 97.9 | 0.56 |
| TQL 200 (expected mean delay: 106.7 ms) | FEM 104.792 20.8714 | PI 136.754 37.9652 | 97.22 | 0.14 |
|                          | ARED 108.91 69.9759    | 97.5 | 0.63 |

Fig. 14. Scenario II-2 (500 flows): PI Queue length with default parameter values

Fig. 15. Scenario II-2 (500 flows): PI Queue length with new parameter values

Fig. 16. Scenario II-2 (500 flows): FEM Queue length
length evolution of the PI controller in the case of Scenario II-2 for 500 TCP flows. The default values of the PI parameters are used, as those given in [13]. It can be seen the poor performance of the PI controller in the case of an increased traffic load and high RTT. It cannot manage the queue, leading to sustain packet losses due to overflow. Also, at the time of the traffic changes, the PI controller shows a sluggish response to maintain the queue at the desired queue length. Thus, we have manually changed the PI parameters, by following the design rules explained in [13] (considering the particular network conditions for this experiment, that is, the increased number of TCP flows, as well as the high RTT), and the new result is shown in Figure 15. As it can be seen, the new values have improved the performance of the PI controller, as compared to the Figure 14, even though the sluggish response and large buffer fluctuations still exist. This experiment demonstrates a major weakness of the PI controller: having fixed parameters that are dependent on network parameters, like the number of flows and round-trip time. Hence, in the case of dynamic, varying conditions the PI controller fails to exhibit a robust behaviour. On the other hand, the fuzzy logic based AQM technique (FEM) shows that is not being influenced by the increased number of TCP flows and RTT delays, and it still exhibits a robust behaviour by successfully maintaining the queue around the equilibrium, resulting in high utilization and minimal losses (see Figure 16).

B. Scenarios III

The network topology used is shown in Figure 17. We have considered network topologies with multiple bottleneck links in order to examine the performance of the AQM schemes in more realistic scenarios. We use AQM in the queues of all core links from router-A to router-F. All other links (access links) have a simple Tail Drop queue. The link capacities and propagation delays are set as follows: \((C_l, d_l) = (C_6, d_6) = (C_9, d_9) = (100\text{Mbps}, 5\text{ms})\), \((C_2, d_2) = (C_4, d_4) = (15\text{Mbps}, 10\text{ms})\), \((C_3, d_3) = (15\text{Mbps}, 60\text{ms})\), \((C_5, d_5) = (15\text{Mbps}, 30\text{ms})\), and \((C_7, d_7) = (C_{10}, d_{10}) = (C_{11}, d_{11}) = (200\text{Mbps}, 5\text{ms})\). \(N_1\) flows end up at destination 1, \(N_2\) flows end up at destination 2, and \(N_3\) flows end up at destination 3. The results show that both bottleneck links (between router-B and C, and between router-D and E) exhibit similar behaviour, as far as the performance comparison is concerned. Therefore, due to lack of space, we have chosen the bottleneck link between router-D and router-E to show the results obtained (more detailed results can be found in [46]). All results (the bottleneck link utilization, the loss rate, and the mean queuing delay with its standard deviation) are summarized in Table 4.

In Scenario III-1 the number of flows is \(N_1 = 100\), \(N_2 = 50\), and \(N_3 = 100\). We also introduce the time-varying dynamics on the network, by stopping half of the flows at time \(t = 40\text{ sec}\) and resuming transmission at \(t = 70\text{ sec}\). In Scenario III-2, we have increased the \(N_1\) flows to 500 flows, in order to examine the performance of the AQM schemes in high traffic load with time-varying dynamics that kept as in Scenario III-1. Finally, in Scenario III-3, we kept the traffic conditions as in Scenario III-2 with an increase of the number of \(N_2\) and \(N_3\) flows to 100 and 200 flows respectively. The expected queuing delay experienced at router-D is 106.7 ms (15Mbps link capacity corresponds to 1875

![Fig. 17. Multiple-bottleneck network topology](image)

| Table 4. Scenarios III: Summary of statistical results |
|---|---|---|---|
| Scenarios | Mean-Delay (ms) | Std-Deviation (ms) | Utilization (based on goodput) (%) | Loss rate (%) |
| III-1 | FEM | 110.47 | 17.39 | 99.8 | 0.08 |
| | PI | 152.42 | 75.57 | 99.2 | 0.68 |
| | ARED | 112.12 | 23.65 | 93.1 | 6.91 |
| III-2 | FEM | 117.42 | 25.76 | 99.67 | 0.32 |
| | PI | 207.71 | 72.73 | 97.47 | 2.68 |
| | ARED | 154.90 | 43.65 | 87.33 | 12.51 |
| III-3 | FEM | 117.77 | 20.50 | 99.87 | 0.34 |
| | PI | 226.33 | 55.49 | 96.73 | 3.86 |
| | ARED | 162.46 | 43.90 | 84.13 | 15.83 |
packets/sec; for a TQL of 200 packets the expected mean delay is 200/1875 = 0.1067).

Figure 18 shows the loss rate as the total traffic load increases, where it can be seen that FEM has the lowest drops (it experiences minimal losses, much below 1%). FEM shows stable and low packet loss over large traffic load. A-RED has the largest drops with a large increase of packet loss with respect to higher loads. Figure 19 shows the utilization of the bottleneck link with respect to the mean queuing delay. FEM outperforms other AQM schemes on both high utilization and low mean delay, thus it exhibits a more stable, and robust behavior, even though the traffic load is increasing. The other AQM schemes show a poor performance as the number of total traffic load increases, achieving much lower link utilization, and large queuing delays, far beyond the expected value. From Table 4, it is clear that FEM has the lowest variance in queuing delay, resulting in a stable and robust behavior. On the other hand, the other AQM schemes exhibit very large queue fluctuations with large amplitude that inevitably deteriorates delay jitter.

We further conducted an experiment to investigate the influence of the traffic load on the behavior of the A-RED controller. As indicated in Section II-B, RED shows a great sensitivity on variation of traffic load, and tries to stabilize the queue as the number of connections increases, but only at the cost of large queues, and consequently increased delay. Figure 20 shows the queue length evolution of A-RED in the case of Scenario III-2, where the total number of TCP flows has been increased significantly from 250 to 650 flows. It can be seen that RED tries to stabilize the queue around 350 packets, just above the maximum
threshold (and far away from the average TQL), having the impact of an increased mean queuing delay and loss rate. At the time of the traffic changes, A-RED shows a good response to maintain the queue at the desired average queue length. However, the global behaviour of A-RED, as shown in Figure 20, is that it is badly influenced by the traffic load. On the other hand, the fuzzy logic based AQM technique (FEM) shows that is not adversely influenced by the increased number of TCP flows, and it still exhibits a robust behaviour by successfully maintaining the queue around the equilibrium, resulting in high utilization and minimal losses (see Figure 21 and Table 4).

IV. Remarks

Some important remarks that resulted from the study done on existing AQM mechanisms follow.

At first, as the RED-based algorithms control the macroscopic behaviour of the queue length (looking at the average) they often cause sluggish response and fluctuation in the instantaneous queue length. As a result, a large variation in end to end delays is observed.

A-RED attempts to tune the RED parameters for a robust behavior but fails to do so in various dynamic cases due to the fact that A-RED retains RED’s basic linear structure. Also, A-RED’s mechanism for adaptation of maximum drop/mark probability is too slow to follow traffic bursts/short-lived flows, as it depends on constant increase/decrease step sizes. This causes a high queue length that can lead into forced drops in the case of the arrival of a traffic burst.

The PI controller behaves in a similar way by exhibiting sluggish response to varying network conditions. This can be explained due to the fact that as the fixed/static PI parameters are dependent on network parameters, like the number of flows and round-trip time, it is difficult to get a stable operation in a broad range of dynamic varying traffic conditions. In addition, the PI controller is based on a simplified linear model which does not consider the slow start phase of TCP and timeout events. These modelling assumptions are not consistent with what real/live measurements indicate [32, 33, 35, 36], and should be taken into account. In particular, the TCP traffic constitutes 90 percent of all traffic with 50-70 percent of this TCP traffic being short-lived connections both in size and in lifetime (the so called mice) [32-33]. In case of short-lived flows, TCP is strongly affected by slow start phases, with segment losses mostly being timeout losses [35]. Also, real TCP traces used in [36], where a stochastic model for the steady-state throughput of long-lived bulk transfer TCP flows is developed, contained more timeout loss events than fast retransmit events. Therefore capturing the effect of the timeout mechanism is important from a modelling prospective. Thus the correctness of the derived model is questioned, when considering today’s Internet dynamics. The PI controller shows difficulties to accommodate itself to the complex, nonlinear, time-varying network status, with bursty, short-lived flows, and also unresponsive flows. This modelling approach is common, due to the difficulty in modelling the effect of slow start and/or timeout events, thus it is questionable whether the proposed models can capture the rich dynamics, and thus making the analysis usefulness questionable.

Moreover, the dynamics of TCP/AQM models are studied with the aid of linearization around equilibrium points of the nonlinear model developed, in order to study TCP/AQM stability around equilibrium. However, linearization fails to track the system trajectories across different regions dictated by the nonlinear equations derived. As stated in [39], linearization “assumes, and hence requires that the system always stays within a certain operating regime”. Furthermore, the equations modeled are dependent on various network parameters, such as the number of flows, and the round trip delays, which in current Internet these parameters vary substantially. Therefore by linearization around specific operating point and the dependence on varying network parameters makes it difficult to get a stable and robust operation, in the case of TCP/IP networks with dynamic load and delay changes.

Hence, a major weakness of the proposed models is that the configuration of control...
parameters is done for a specific operating point for which various system’s parameters are assumed to be known, or certain dynamics ignored. As stated in [41], even if the assumptions regarding the input parameters fit the specific scenario, the applicability of the AQM algorithm would be restricted to a small range of the assumed values only. Therefore, the configured parameter set and stability conditions introduced by the proposed models lack applicability to all possible real scenarios with varying dynamics of network conditions.

In addition, even if the linearized system is made stable at equilibrium, there is no guarantee that the nonlinear system will remain stable [41], especially if the deviations from the equilibrium are at times large. As stated in [38], instability is undesirable, and can cause three problems: (i) it increases variations in source rate and delay, (ii) it subjects short-lived transfers to unnecessary delay and loss, and (iii) it can lead to underutilization of network links if queues bound between empty and full.

The main result of this study is, therefore, that by using nonlinear drop/mark probability function an effective and robust AQM system can be designed to drive quickly the system to be controlled into the steady-state, as opposed to a linear drop/mark probability function that itself is not robust enough for the highly bursty network traffic and cannot capture the dynamics and nonlinearities of the controlled system. For example, as observed in [41], during high load condition a disproportionately higher drop/mark probability is required than in a low load condition, in order to keep the queue length in the same range, a requirement met only by a nonlinear drop/mark function.

Thus, the complexity of these problems and the difficulties in implementing conventional controllers to eliminate those problems motivate the need to investigate intelligent control techniques such as fuzzy logic as a solution to controlling systems in which time delays, nonlinearities need to be addressed. The proposed nonlinear fuzzy logic based AQM technique is shown, via simulative evaluation, to exhibit many desirable properties, like robustness and fast system response, with capabilities of adapting to highly variability and uncertainty in network.

Of course, fuzzy logic control has its own limitations. Much work remains for the analytical study of fuzzy logic, particularly in the area of stability and performance analysis. Most proposed fuzzy logic controllers in literature do not have any stability analysis because of the difficulty in analysis. This is mainly due to the existence of nonlinearity in the control structure that usually makes it difficult to conduct theoretical analysis to explain why fuzzy logic controllers can achieve better performance than the conventional counterparts.

V. Conclusions and Future Work

This paper presents an overview of a number of AQM mechanisms in TCP/IP networks, discusses the modelling and control approach that is followed and addresses possible limitations. Such limitations identified are:

- The dependency of AQM control parameters on dynamic network parameters, like the number of flows and the round trip propagation delays.
- The accuracy of the proposed TCP/AQM models, as they ignore the slow start phase of TCP and/or timeout events that are prominent conditions in today’s Internet with the existence of short-lived TCP/Web flows.
- The linearity of the control functions of proposed AQM mechanisms that cannot capture effectively the nonlinearities of the TCP network.

Emphasis is given towards the ability of effectively controlling the congestion in dynamic, time-varying TCP/IP networks.

From the results, it is evident that linear heuristic-based mechanisms (e.g., A-RED [12]) and linear control theory based techniques (e.g., PI [13]) still require careful configuration of their control parameters. They are often non-robust to dynamic network changes, and as a result, they exhibit greater delays than the target mean queuing delay and large buffer.

\footnote{In this paper we use a nonlinear fuzzy logic based technique to derive a nonlinear drop/mark probability function}
fluctuations (delay variation), and consequently cannot effectively control the router queue.

On the other hand, the nonlinear control structure developed in FEM controller, using a fuzzy logic based technique, is shown by extensive simulations [4, 5, 6, 7, 46] to be able to compensate for varying round trip delays and number of active flows, and show significant improvement in maintaining performance and stability over a wide range of operating conditions, in contrast with difficulties appearing when using analytical approaches (e.g., PI [13]). Fuzzy logic control methodology shows the potential of incorporating human knowledge into such a control strategy. The capability to qualitatively capture the attributes of a control system based on observable phenomena is a main feature of fuzzy logic control and has been demonstrated from the simulative evaluation in [4, 5, 6, 7, 46]. The proposed Fuzzy Control methodology is expected to offer significant improvements on controlling congestion in TCP/IP networks and thus providing quality of service.

In overall, in the complex, but challenging, concept of TCP/AQM, many issues remain to be investigated.

Firstly, proposed AQM mechanisms and their derived models should be validated concerning a wide range of dynamic network conditions that fit the real environment requirements in today’s Internet. These include heterogeneous TCP traffic such as flows with different round trip propagation delays and a mixture of short and long lived flows, as well as considering multiple bottleneck links. The validation should also consider new TCP implementations such as TCP SACK, and other traffic types like open-loop (uncontrolled) UDP traffic (such as voice) and mixes of TCP and UDP traffic.

In addition, new models should include the slow start phase of TCP and timeout events in order to accurately capture today’s Internet dynamics. In order to demonstrate their merits, new models need to be compared to existing models based on a common set of traffic measurements and/or validated using live measurements. The (adequate) understanding of the TCP/AQM dynamics and an accurate enough modeling can significantly enhance the prediction capabilities, leading to better quality of service and better utilization of networking resources.

Furthermore, from the experience gained with the use of fuzzy logic, the development of nonlinear control function that best captures the nonlinearities of the controlled system can be further investigated, in order to get a stable and robust operation of a TCP/AQM system.

Finally, a measure of stability or a certain degree of safety concerning fuzzy logic based AQM can be examined.

REFERENCES


