

Article

Diffusion Model of a Non-Integer Order PI^γ Controller with TCP/UDP Streams

Dariusz Marek ¹ , Adam Domański ¹ , Joanna Domańska ² , Jakub Szygula ^{1,*} , Tadeusz Czachórski ²  and Jerzy Klamka ² 

¹ Department of Distributed Systems and Informatic Devices, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland; dariusz.marek@polsl.pl (D.M.); adam.domanski@polsl.pl (A.D.)

² Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland; joanna@iitis.pl (J.D.); tadek@iitis.pl (T.C.); jerzy.klamka@iitis.pl (J.K.)

* Correspondence: jakub.szygula@polsl.pl

Abstract: In this article, a way to employ the diffusion approximation to model interplay between TCP and UDP flows is presented. In order to control traffic congestion, an environment of IP routers applying AQM (Active Queue Management) algorithms has been introduced. Furthermore, the impact of the fractional controller PI^γ and its parameters on the transport protocols is investigated. The controller has been elaborated in accordance with the control theory. The TCP and UDP flows are transmitted simultaneously and are mutually independent. Only the TCP is controlled by the AQM algorithm. Our diffusion model allows a single TCP or UDP flow to start or end at any time, which distinguishes it from those previously described in the literature.

Keywords: active queue management; diffusion approximation; fractional controller PI^γ ; internet; TCP/IP and UDP



Citation: Marek, D.; Domański, A.; Domańska, J.; Szygula, J.; Czachórski, T.; Klamka, J. Diffusion Model of a Non-Integer Order PI^γ Controller with TCP/UDP Streams. *Entropy* **2021**, *23*, 619. <https://doi.org/10.3390/e23050619>

Academic Editor: José A. Tenreiro Machado

Received: 24 April 2021

Accepted: 13 May 2021

Published: 16 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Network protocols control the transfer of information between a transmitter and receiver. They have to fulfil many functionalities, for example, ensure the correctness of transmitted data and increase channel capacity. This article focuses on the problem of transmission efficiency. Traffic intensity has a stochastic nature. The protocols should adapt the transmission to the limited bandwidth and deterministic mechanisms of computer networks. The goal is to minimize information loss due to packet drops in overflowing buffers. These control activities are based on the exchange of information between sender and receiver. The article analyses two approaches to increase the efficiency (by minimizing packet loss) in the wide-area network:

- a mechanism for controlling the speed of sending information at the transmitter level, due to the TCP protocol [1];
- removing data from the network regulated by the AQM mechanism on the IP level [2].

Due to the rapid development of network communication technology, more and more attention has been focused on the problems of congestion control. Congestion is an important factor affecting a network's Quality of Service (QoS) and reducing its performance. The Transport Control Protocol (TCP) has already been used in Internet applications for more than 30 years. During this time, several congestion control algorithms have been developed to meet the requirements of a constantly changing computer network. Active Queue Management (AQM) is a network approach to congestion prevention that works in combination with the TCP protocol. The most effective congestion control occurs when the AQM mechanism and TCP protocol work together [3].

The earliest of the AQM algorithms is called Random Early Detection (RED) [4]. In the past twenty years, many AQM mechanisms have been proposed. These mechanisms

can be classified into three categories [5,6]: heuristic, optimization and control theory approach. The heuristic approach towards the AQM mechanism heavily depends on intuition. Examples of such mechanisms are inter alia, BLUE algorithm [7], hyperbola RED (HRED) [8], and Yellow algorithm [9]. These algorithms aim to improve packet loss, fairness, network utilization, and adaptability to different characteristics of the network traffic. The AQM based on the optimization approach was developed by Frank Kelly. His paper [10] transforms the design of the AQM algorithm to a convex optimization problem. Another AQM algorithm elaborated following the optimization theory was the Random Early Marking (REM) [11]. However, the AQM mechanisms based on the optimization theory are usually complex and hard to tune [6]. In this case, it is also difficult to control the instantaneous queue length of the router [12]. To overcome these effects, researchers have resorted to the control theory. The exact type of a PI^γ controller used in this article belongs to the family of AQM mechanisms based on this theory, described in Section 2.

The dynamic model of the TCP behavior is required to enable the application of the control theory principles to AQM. The transient analysis should be performed in the model to provide us with time-dependent behavior of flows and queues. Apart from discrete time simulation, which is very time-consuming in the case of transient-state analysis, fluid flow or diffusion approximation methods can be used. Usually, due to its simplicity, the fluid flow method is the one used to model TCP networks [13–16]. The diffusion approximation, on the other hand, offers more accurate results. In the diffusion approximation model, traffic flows are determined by their mean and variance. On top of that, second-order partial equations describe queue changes.

In this paper, the transient behavior of the AQM mechanisms is analyzed via extending our earlier models presented in [17]. Comparing with previous results, the consideration of independent TCP/UDP streams that can start and end at any time constitutes a novelty. The diffusion approximation method is employed to trace the behavior of the AQM mechanisms applied to control Internet traffic. We are able to describe the features of RED, NLRED, PI and PI^γ that were not possible to notice in an open-loop or a fluid-flow scenario. Table 1 presents main approaches to AQM based on PI and non-integer order PI^γ controllers.

Table 1. Main approaches of AQM based on PI and non-integer order PI^γ controllers.

PI (Simulation) [18]	Study of the TCP/AQM mechanisms based on PI controllers
PID (Simulation) [19]	Evaluation of the AQM based on non-integer order PID controller
PI^γ (Fluid-Flow) [20]	First application of non-integer order PI^γ controller to an AQM strategy
PI^γ (Fluid-Flow/Simulation) [21]	Fluid flow approximation and discrete-event simulation to investigate the influence of the AQM policy based on non-integer order PI^γ controller on the packet loss probability, the queue length and its variability
$PI^\gamma D^\omega$ (Simulation) [22]	Model of AQM mechanism based on non-integer order $PI^\gamma D^\omega$ controller
PI^γ (Simulation) [23]	Finding optimal parameters of the non-integer order PI^γ controller used as AQM mechanisms. The optimization was made by using the well-known Hooke and Jeeves direct search method applied for minimization of a multivariate score function
Adapted PI^γ (Simulation) [24]	Choice of non-integer order PI^γ controller parameters based on machine learning algorithms. The controller parameters automatically adjust to network traffic parameters (traffic intensity and self-similarity)
TCP PI^γ (Diffusion) [17]	The diffusion approximation model of the simple TCP traffic. Evaluation (in close loop scenario) of the effectiveness of active queue management (AQM) mechanisms based non-integer order PI^γ controller
TCP PI^γ (Combined Diffusion and Simulation) [25]	Combined diffusion approximation and simulation model based on non-integer order PI^γ controller

This paper is organized as follows: In Section 2 one can find the literature overview of AQM mechanisms based on the control theory approach. This section also describes an embedding of our model in the area of diffusion modelling. Section 3 gives a brief description of the AQM mechanism used in this paper. Section 4 describes our diffusion

model of the PI^γ controller with TCP/UDP streams. In Section 5, numerical results are presented. Section 6 concludes our work.

2. Background and Related Work

2.1. AQMs Based on the Control Theory Approach

Yes, it should be subsection, thank you. The detection and mitigation of congestion are some of the biggest problems in the computer networks domain [3]. Many mechanisms have been created to solve these issues. They are based on different operating principles. Most of the congestion avoidance mechanisms have been implemented in transport layer protocols such as TCP [26]. UDP applications, such as voice or video traffic, do not have any congestion avoidance mechanisms. In the case of such applications, it is possible to use the open-loop hop-by-hop backpressure strategy. In accordance with this strategy, a congested node broadcasts backpressure messages to upstream nodes to reduce their transmission rate [27]. This method is useful for bandwidth control but, unfortunately, introduces transmission delays. Therefore, there are many attempts to improve these kinds of algorithms, for example, [28]. Back pressure algorithms can be used in wireless sensor networks (WSNs) because of their limited need for: computation resources, storage, energy and communication bandwidth [27,29]. The TCP protocol and the back pressure mechanism are incompatible due to a mismatch between the TCP congestion control mechanism and the back pressure queue size based routing [26]. As a result, wider use of this solution in the Internet would require changes in the TCP protocol [26].

The traffic control in TCP/IP network is, in fact, a closed-loop algorithm in which AQM algorithm plays the role of a controller. The performance and dynamics of network connections can be studied using the control theory to improve their stability and to reduce the reaction time. Several feedback control algorithms have been developed. The article [30] proposes a fluid flow dynamic model of TCP/RED networks by using stochastic differential equations. Based on this dynamic model, several AQM controllers have been proposed using different control approaches. The article [13] proposes a Proportional-Integral (PI) controller on low-frequency dynamics. Authors of the article [31] propose adaptive Proportional (P) and Proportional-Integral (PI) controller and conclude that PI controller can adapt very well to the large fluctuation of the Internet traffic. The article [32] describes a new variant of the RED mechanism called Proportional-Derivative-RED (PD-RED) that performs better than Adaptive RED. Authors of [19] propose the Proportional-Integral-Differential (PID) controller to speed up the responsiveness of the AQM system. Among them, PI controllers are attracting increased attention because of their computation, and implementation simplicity [18]. The article [6] tries to preserve the simplicity of the PI controller by proposing a self-tuning compensated PID controller.

Traditional calculus is based on integer-order differentiation and integration. Differentiation or integration of non-integer order have been used in many mathematical models of dynamic systems. The article [33] claims that many real dynamic systems are better characterized using a fractional dynamic model. Authors indicate that non-integer order controllers provide better performance than the conventional integer order ones. The article [20] presents the first application of the fractional order PI controller to an AQM strategy. The authors focus on the method for determining the parameter regions where the PI^γ controller ensures a given modulus margin (inverse of the H_∞ norm of the sensitivity function). The article [21] describes an evaluation of the fractional-order PI^γ controller used as an AQM mechanism. The performance of the controller is evaluated using fluid flow approximation (closed-loop control) and simulation (open loop scenario). The article [34] studies the proper selection of the $PI^\alpha D^\beta$ parameters to show an influence of the proportional, integral and derivative terms on the controller's dropping function. A simulation model is used in this article. The article [17] proposes a new model of the PI^γ controller based on a diffusion approximation approach. This model is able to provide more detailed information on transmission delays than the frequently used fluid flow model.

2.2. Diffusion Approximation

Diffusion models refer to the changes of flows defined by their mean and variance. They are more accurate than the fluid flow models, where only the mean value is considered. They use the central limit theorem to justify that the number of arrivals and services at a queueing system tends to the normal distribution and a diffusion process may represent the queue length. The solution of the diffusion equation with parameters depending on the analyzed system approximates a queue distribution. This way, the models easily incorporate general distributions of interarrival and service times and the transient queue behavior.

The diffusion approximation has been used to study the performance evaluation of computer systems and networks for many years. The tutorial [35] describes how the diffusion approximation formalism can be applied to the analysis of some traffic control mechanisms in the ATM network. The article [36] presents the diffusion model of wireless network based on the IEEE 802.11 protocol. In the article [37], the first attempt to model the TCP/RED router using diffusion approximation is made. The article [38] applies diffusion approximation to model the influence of a buffer capacity on Quality of Experience in wireless video connections. In our article [17], a new model of TCP NewReno based on the diffusion approximation method is developed. The combined diffusion approximation and simulation model is proposed in [25]. These models allow analyzing the behavior of a single TCP stream. To the best of our knowledge, there is no diffusion model described in the literature which can be used to analyze the independent TCP/UDP streams.

3. RED, NLRED and a Non-Integer Order PI^γ Controllers

The traffic control built-in a TCP/IP protocol is a typical closed-loop one. The flow of packets emitted by a sender is controlled by the loss probability p observed in routers and reported to the sender with a certain delay. The losses decrease the traffic, and their lack increases it. In the case of a passive router, the losses occur when the router queue is full.

The RED family algorithms determine the dropping probability even if there is still a place to store packets but the queue increases. They use the weighted moving average avg computed at packets' arrival; for packet i :

$$avg_i = (1 - w)avg_{i-1} + wq_i$$

where avg_{i-1} is the moving average computed at the arrival of the previous packet, and q_i is the queue seen by the packet i . The probability of a loss is for small values of avg $p_{RED} = 0$. As avg grows, the probability increases linearly between two thresholds Min_{th} and Max_{th} , from 0 to p_{max} .

$$P_{RED}(avg) = p_{max} \frac{avg - Min_{th}}{Max_{th} - Min_{th}} \quad (1)$$

Finally, it becomes $p_{RED} = 1$ given $avg > Max_{th}$.

Another modification of RED is NLRED (non-linear RED) algorithm [39]. In this mechanism, the linear packet dropping function is replaced by a quadratic function:

$$p = \begin{cases} 0 & \text{for } avg < Min_{th} \\ \left(\frac{avg - Min_{th}}{Max_{th} - Min_{th}} \right)^2 P_{max} & \text{for } Min_{th} \leq avg \leq Max_{th} \\ 1 & \text{for } avg > Max_{th} \end{cases} \quad (2)$$

Paper [40] proposes another approach to a non-linear packet dropping function. This function is based on the third-degree polynomials instead of the well-known quadratic function. This approach allows to choose the optimal packet dropping function:

$$p(avg, a_1, a_2, p_{max}) = \begin{cases} 0 & \text{for } avg < Min_{th} \\ \varphi_0(avg) + a_1\varphi_1(avg) + a_2\varphi_2(avg) & \text{for } Min_{th} \leq avg \leq Max_{th} \\ 1 & \text{for } avg > Max_{th} \end{cases} \quad (3)$$

where:

$$\varphi_0(avg) = p_{max} \frac{avg - Min_{th}}{Max_{th} - Min_{th}}, \quad (4)$$

$$\varphi_1(avg) = (avg - Min_{th})(Max_{th} - avg), \quad (5)$$

$$\varphi_2(avg) = (avg - Min_{th})^2 (Max_{th} - avg) \quad (6)$$

Probability p can also be determined directly by the controller by comparing the current queue q_i and the queue q_0 we wish to maintain. Their difference, called error in the control theory, is the input signal to the controller, which may be the classical PID one or, as it has been investigated recently, the proportional-integral PI^γ of fractional, that is, non-integer order. Fractional derivatives and integrals are known since the times of Leibnitz and recently become used in the control of physical processes [41].

In the case of the PI^γ controller, the loss probability p_i of a packet i is equal to:

$$p_i = \max\{0, -(K_P e_i + K_I \Delta^\gamma e_i)\} \quad (7)$$

It depends on the proportional and integral terms K_P, K_I , the error $e_i = q_i - q$, and the order of integration γ . Their impact is further discussed in the articles [21,22,34,42].

The packet drop probability is determined at discrete moments of packet arrivals. There exists only one definition of the non-integer order discrete differ-integral. It [43] is a generalization of the traditional definition of the difference between the integer-order and the non-integer one and is analogous with the generalization employed in the Grünwald-Letnikov (GrLET) formula [44,45].

For a given sequence $f_0, f_1, \dots, f_j, \dots, f_k$

$$\Delta^\gamma f_k = \sum_{j=0}^k (-1)^j \binom{\gamma}{j} f_{k-j} \quad (8)$$

where $\gamma \in R$ is by and large a non-integer fractional order, f_k is a differentiated discrete function, and $\binom{\gamma}{j}$ is a generalised Newton symbol which definition looks as follows:

$$\binom{\gamma}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \frac{\gamma(\gamma-1)(\gamma-2)\dots(\gamma-j+1)}{j!} & \text{for } j = 1, 2, \dots \end{cases} \quad (9)$$

Parameters of the non-integer order PI^γ controller are presented in Table 2.

Table 2. Parameters of the non-integer order PI^γ controller.

K_P	Proportional term
K_I	Integral term
γ	Integral order
e_i	Error in current slot
q_i	Actual queue length
q	Desired queue length

4. Diffusion Approximation of the TCP and UDP Network Streams

In this section, how to model the AQM router supporting TCP/UDP flows using the diffusion approximation is described. The main goal of the analysis presented below is to model the Active Queue Management based on the answer of PI^γ controller.

The method of diffusion approximation is used in queueing theory (e.g., [46–49]) when it is hard to determine a queue distribution. The queue length is replaced by the value of diffusion process $X(t)$. The probability density function (pdf) of the letter, $f(x, t; x_0)$

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx | X(0) = x_0] \quad (10)$$

of $X(t)$, is given by the diffusion equation:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \quad (11)$$

and helps us to evaluate the queue distribution. It is usually used in the analysis of G/G/1 queueing systems, that is, having general distributions of interarrival and service times or G/G/1/L, where additionally the queue is limited to L positions. In the latter case the number of packets in router is in the range $[0, L]$ and therefore the diffusion process is limited by barriers in $x = 0$ and $x = L$, and the diffusion equation is used in the form [47]:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} = & \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \\ & + \lambda p_0(t) \delta(x - 1) + \lambda p_L(t) \delta(x - L + 1) \end{aligned}$$

$$\begin{aligned} \frac{dp_0(t)}{dt} = & \lim_{x \rightarrow 0} \left[\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda p_0(t), \\ \frac{dp_L(t)}{dt} = & \lim_{x \rightarrow L} \left[\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \mu p_L(t). \end{aligned} \quad (12)$$

In the above equations, $[p_0(t), p_L(t)]$ denote probabilities that the process is in either of barriers, λ and μ are the intensities of jumps from barriers; from $x = 0$ to $x = 1$ and from $x = L$ to $x = L - 1$ corresponding to the arrival of a packet to the empty queue or departure of a packet from the full queue. Parameters β and α are chosen as $\beta = \lambda - \mu$, $\alpha = \lambda^3 \sigma_A^2 + \mu^3 \sigma_B^2$, where $1/\lambda, 1/\mu$ are the first moments of interarrival and service time distributions, and σ_A^2, σ_B^2 are their variances, more information please see Table 3. This way the changes of the diffusion process and of the queue have the same mean and variance. The steady-state solution of Equation (12) is given in [47] and the transient case is considered in [49].

Table 3. Main notations and parameters of the diffusion model.

λ	Intensity of the input traffic
μ	Intensity of packet processing and dispatching
σ_A^2	Variance of interarrival time distribution
σ_B^2	Variance of service time distribution
C_A^2	Squared coefficient of variation of interarrival time distribution
C_B^2	Squared coefficient of variation of service time distribution
$X(t)$	Diffusion process
β	Diffusion parameter; βdt is the mean value of changes of $X(t)$ during dt
α	Diffusion parameter; αdt is the variance of changes of $X(t)$ during dt
$f(x, t, x_0)$	Probability density that the process will be in state x at time t , for initial conditions x_0

Below, we expand this model to include a number of independent streams. We assume a queue supports k input streams. For each stream, packets arrive at intervals which are described by the distribution $A^{(k)}(x)$. The service time distribution is equal to $B^{(k)}(x)$. The k input stream ($k = 1, \dots, K$) is described by the distribution $A^{(k)}(x)$, (with the average value $1/\lambda^{(k)}$ and variance $\sigma_A^{(k)2}$). The service time of the k stream has a distribution $B^{(k)}(x)$ with the average value $1/\mu^{(k)}$ and variance $\sigma_B^{(k)2}$. The density functions of these distributions are denoted by $a^{(k)}(x)$ and $b^{(k)}(x)$. We assume input streams of individual

classes are independent. The normal distribution of the number of packets of a k -th stream coming over a period of time is approximately equal to $\lambda^{(k)}t$, with a variance satisfying the equation $\lambda^{(k)3}\sigma_A^{(k)2}t = \lambda^{(k)}C_A^{(k)2}t$. The number of packets of all streams that arrived during this time has also a normal distribution with the average value $\lambda t = \sum_{k=1}^K \lambda^{(k)}t$ and a variance satisfying the equation $\lambda C_A^2 t = \sum_{k=1}^K \lambda^{(k)}C_A^{(k)2}t$. So, the parameters of the totality of streams are:

$$\lambda(t) = \sum_{k=1}^K \lambda^{(k)}(t), \quad C_A^2 = \sum_{k=1}^K \frac{\lambda^{(k)}(t)}{\lambda(t)} C_A^{(k)2} \quad (13)$$

where $\frac{\lambda^{(k)}(t)}{\lambda(t)}$ is the probability that a given packet belongs to a stream k which allows us to determine the resultant service time parameters:

$$\frac{1}{\mu} = \sum_{k=1}^K \frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)}}, \quad C_B^2 = \mu^2 \sum_{k=1}^K \left[\frac{\lambda^{(k)}}{\lambda} \frac{1}{\mu^{(k)2}} (C_B^{(k)2} + 1) \right] - 1, \quad (14)$$

and then the parameters α, β of the diffusion equations:

$$\beta(t) = \lambda(t) + \mu, \quad \alpha(t) = \lambda(t)C_A^2 + \mu C_B^2$$

The distribution $p(n) \approx f(n)$ specifies the number of packets of all streams in the queue, and the probability that there are v packets which belong to the k -th stream in the queue equals to:

$$p^{(k)}(v) = \sum_{n=v}^{\infty} [p(n) \binom{n}{v} \left(\frac{\lambda^{(k)}}{\lambda} \right)^v \left(1 - \frac{\lambda^{(k)}}{\lambda} \right)^{n-v}], \quad (15)$$

where $k = 1, \dots, K$.

In the article, the case of two kinds of input streams is presented. UDP stream is a CBR stream with assumed number of packets sent per time unit ($\lambda_{UDP}(t)$ is constant) and TCP stream for which the input intensity changes according to the TCP NewReno congestion control algorithm ($\lambda_{TCP}(t)$ is shaped by the AQM mechanism). When the diffusion model is considered, the total intensity of the input stream is equal to the sum of intensities of the components.

In the case of two input streams, one TCP and one UDP, the mean and the variance of input distribution for a queue is calculated as follows:

$$\begin{aligned} \beta(t) &= \lambda_{TCP}(t) + \lambda_{UDP}(t) + \mu \\ \alpha(t) &= [\lambda_{TCP}(t) + \lambda_{UDP}(t)] \\ &\quad \left[\frac{\sigma_{A_{TCP}}^2(t)\lambda_{TCP}^3(t) + \sigma_{A_{UDP}}^2(t)\lambda_{UDP}^3(t)}{\lambda_{TCP}(t) + \lambda_{UDP}(t)} \right] + \mu C_B^2 \\ \alpha(t) &= \sigma_{A_{TCP}}^2(t)\lambda_{TCP}^3(t) + \sigma_{A_{UDP}}^2(t)\lambda_{UDP}^3(t) + \mu C_B^2 \end{aligned} \quad (16)$$

The TCP NewReno/AQM model based on the diffusion approximation works as follows: The diffusion approximation gives the distribution of the router's queue at time t . The mean value of this queue length modifies the packet rejection probability. This probability affects the intensity λ of the input stream because the congestion window increases by one for lossless transmission or halves for packet loss. We assume that the flow λ , the size of the congestion window W , and the delay q/μ of the router are linked together in the following way:

$$\lambda = \frac{W\mu}{q} \quad (17)$$

where $1/\mu$ is the mean service time. In the diffusion model, the calculations are carried out inside the intervals of length $\Delta t = 1/\lambda$. When λ changes, the length of the interval changes as well. We obtain a queue distribution at time t_i at the end of the interval i . Based on it, we get the average queue length $E[q_i]$ and then the probability of packet rejection p_i which defines new value of λ : $\lambda_{i+1} = \lambda_i + \Delta\lambda_i$ due to the AQM algorithm, where

$$\Delta\lambda_i = \frac{\mu_i}{E[q_i]} - \frac{\lambda_i^2}{2} \frac{E[q_i]}{\mu} p_i. \quad (18)$$

Over the time $t_{i+1} = t_i + 1/\lambda_i$, we repeat the calculations for a new value of λ . In the model presented in this article it is assumed that the i -th flow can start or end a transmission at any moment. The change in the source intensity $\Delta\lambda$ in the time t affects the dispatching time and the queue length. The algorithm of these calculations is presented in Figure 1.

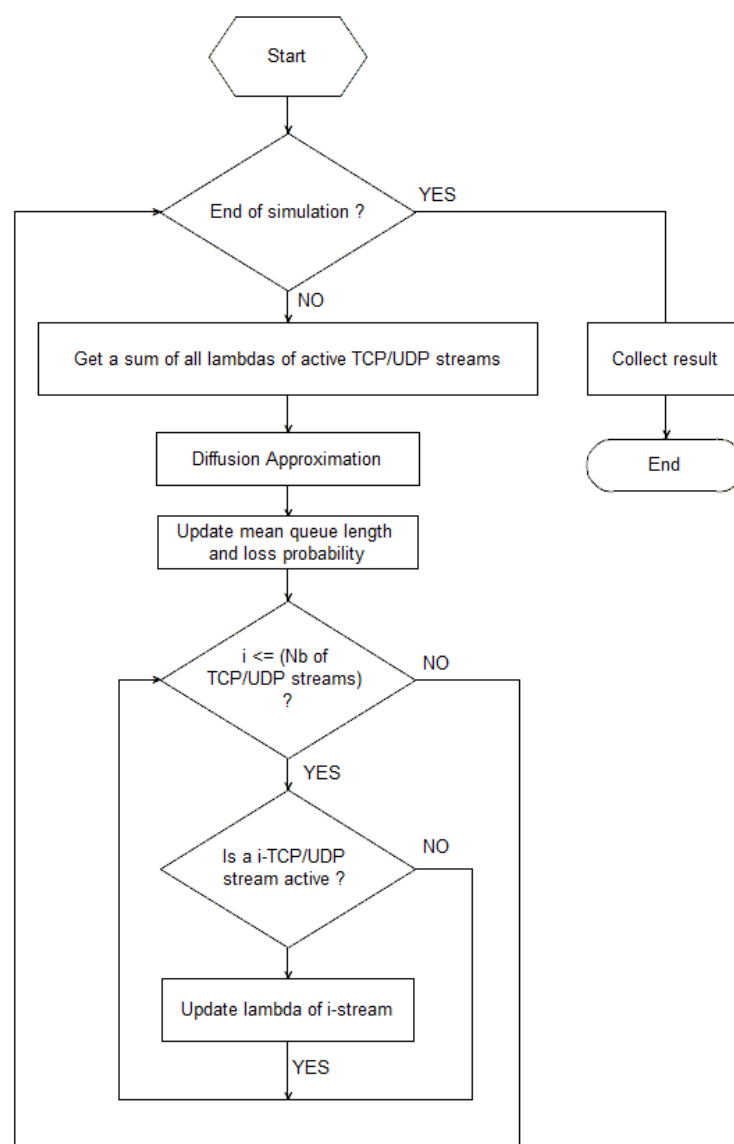


Figure 1. Flowchart of our model behavior.

5. Numerical Results

In this chapter, the results obtained through the method described above are discussed. We employ the proposed numerical scheme to obtain the transient behavior of TCP window

dynamics and queue length. The results for various types of input streams and AQM mechanisms (RED, NLRED, PI, PI^γ) are discussed.

We assume the following parameters of the AQM buffer: $Min_{th} = 10$, $Max_{th} = 20$, buffer size (measured in packets) = 30.

The parameters of RED are as follows: weight parameter $w = 0.008$, $p_{max} = 0.02$. Article [50] shows the impact of RED parameters on the network traffic. Using higher values of these parameters results in the increase in network traffic fluctuations. This choice of RED parameters shows more clearly the influence of TCP/UDP flows on the AQM queue behavior [17]. These parameters are slightly different than those proposed in the literature [51]. The NLRED parameters are as follows: $a_1 = 0.00042$, $a_2 = -0.0000038$ and $p_{max} = 0.855$. Article [40] proposes using parameters with such values to achieve the best transmission performance.

The parameters of PI^γ : $setpoint = 10$, $P = 0.0001$, $I = 0.005$, $\gamma = -0.4$. For $\gamma = -1.0$ the PI controller is considered (Table 4). The proper choice of AQM/PI controller parameters is difficult. It strongly affects the packet dropping function (i.e., integral order γ accelerates and strengthens the controller's response). Proper selection of AQM parameters should help achieve two goals: obtaining desired queue behavior and adaptation to network transmission conditions. The influence of the controller's parameters on queue behavior is discussed in papers [21,22].

Figures 2–17 show the evolution of sources intensity and queue length. In Figure 2 the queue behavior in the case of one TCP flow and PI^γ controller is shown. The source intensity increases until the queue reaches the desired length. Exceeding the desired queue length increases the probability of a packet loss in AQM and thus causes reduction in the source intensity. Consequently, the queue occupancy and packet loss probability decreases. So, the source intensity increases after some time. Source intensity and queue length oscillations continue until a stable state is reached. In the stable state, the average queue length reaches 10.2 packets, and the source intensity reaches $\lambda = 1$. In the case of a PI controller (Figure 2) the average queue length reaches almost the same value (10.01 packets). Nevertheless, the queue occupancy still oscillates between extreme values. Such oscillations, associated with a big fractional order of the controller, are described in the article [24].

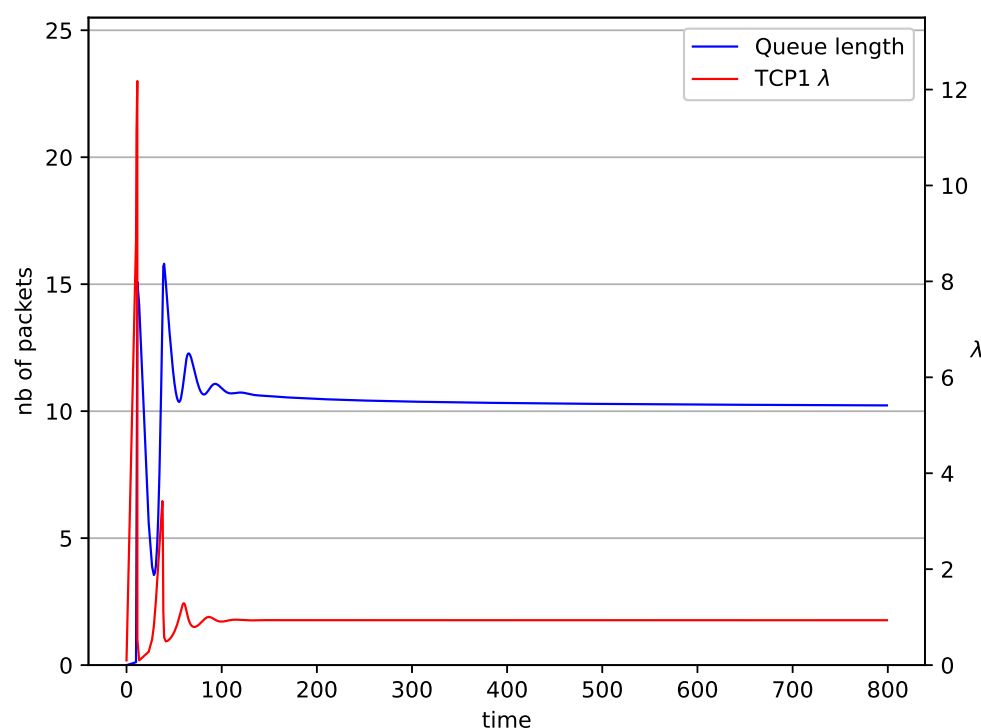


Figure 2. 1 TCP with PI^γ controller.

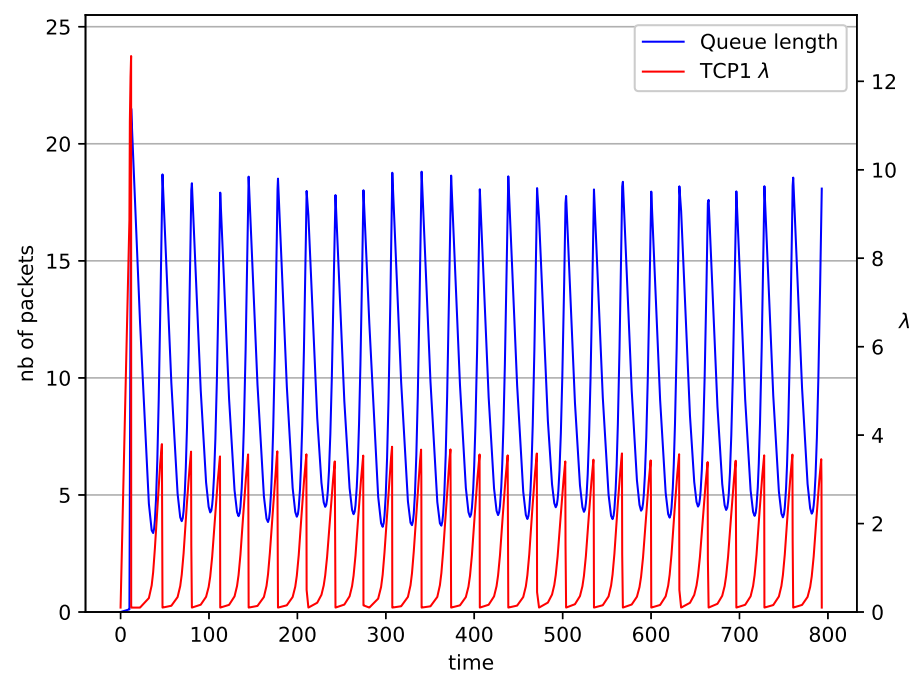


Figure 3. 1 TCP with PI controller.

Table 4. PI^γ and PI controllers coefficients.

	K_p	K_i	γ	Setpoint	Type of Controller
1	0.0001	0.005	−0.4	10	non integer order controller
2	0.0001	0.005	−1.0	10	classical controller

The behaviors of a single TCP stream in cooperation with the RED algorithm (Figure 4) and NLRED (Figure 5) are shown. It is identical to the previous one (Figure 2). The average queue length reaches 14.7 for RED and 10.5 packets for NLRED, the source intensity reaches $\lambda = 1$. This value is roughly halfway between Min_{th} and Max_{th} .

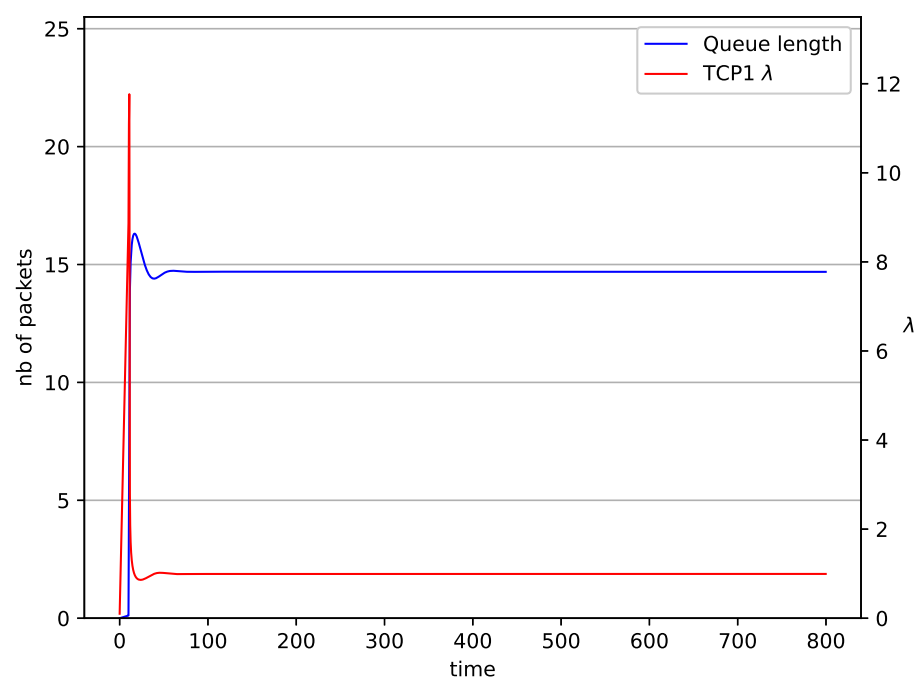


Figure 4. 1 TCP with RED controller.

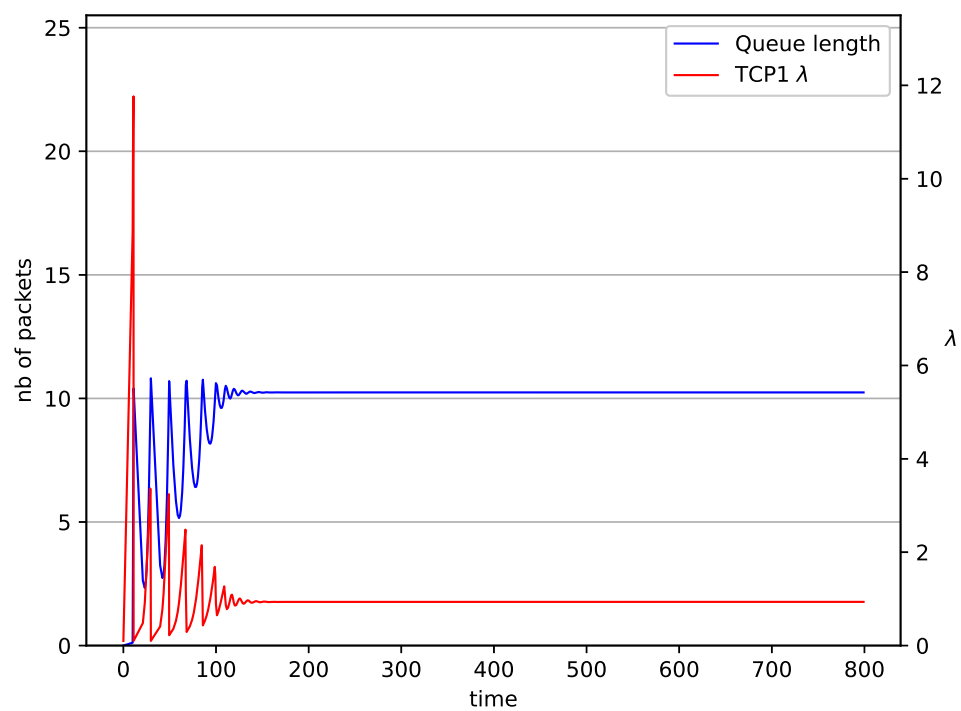


Figure 5. 1 TCP with NLRED controller.

In Figures 6–17 the behavior of the TCP/UDP AQM system for several co-transmitted data streams is portrayed.

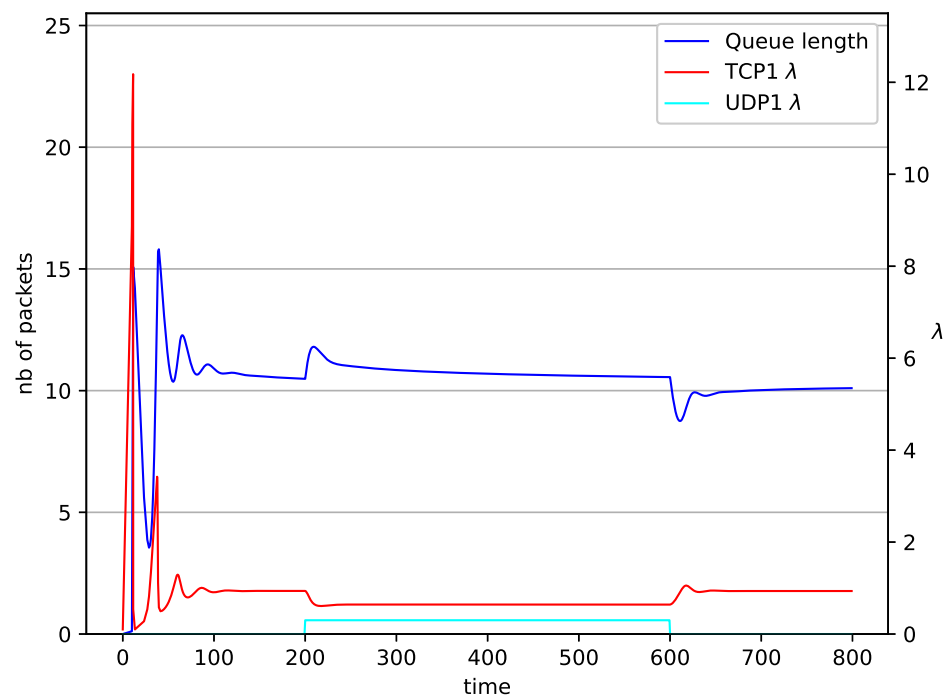


Figure 6. 1 TCP and 1 UDP with PI^γ controller.

In Figures 6–9 simultaneous transmission of TCP and UDP streams is presented. Figure 6 shows the situation where a queue is controlled by the PI^γ mechanism. The TCP transmitter starts a transmission at time $t = 0$. The TCP window evolution proceeds, as shown in Figure 2. At $t = 200$ the UDP stream (constant intensity $\lambda = 0.5$) starts its transmission. Thereby, the intensity of the TCP stream is reduced and the queue occupancy is increased. Over time, the controller attempts to stabilize the queue at the desired level. At $t = 650$, the UDP

stream ends its transmission. As a result, the queue length decreases. The end of the UDP transmission also causes the TCP window to grow. Consequently, the queue length increases up to the desired size. When it comes to the *PI* controller (Figure 7), queue fluctuations occur as previously. They disappear when an additional UDP stream is loaded into the system and reappear once it has finished the transmission. This behavior is also visible in the next experiments. A similar layout of input streams is presented in Figures 8 and 9. In the case of the RED and NLRED algorithms, the commencement of the UDP stream increases the queue length. For the RED mechanism, it reaches the value near Max_{th} .

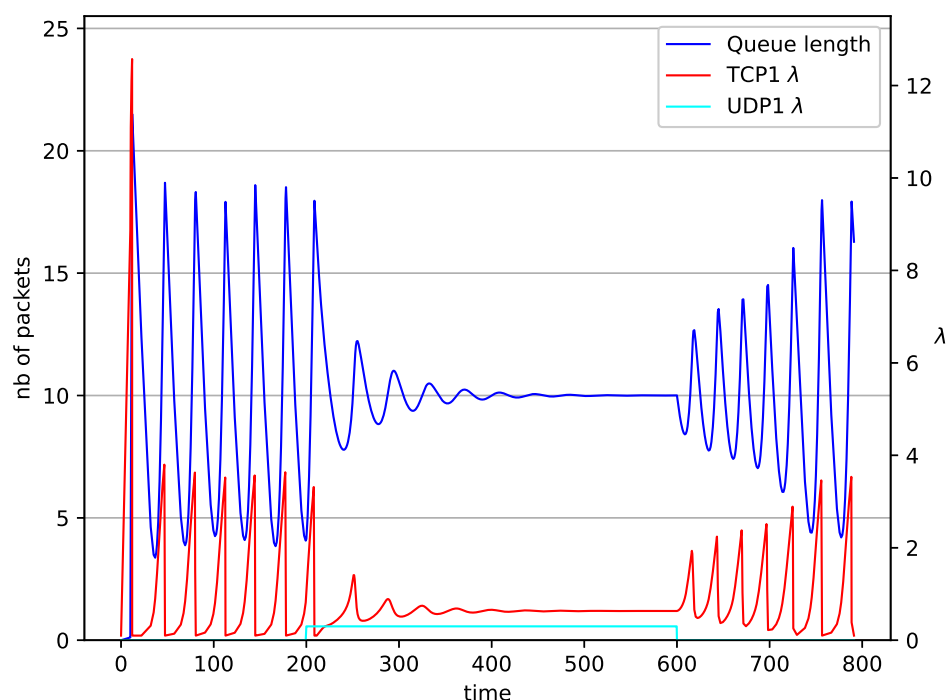


Figure 7. 1 TCP and 1 UDP with *PI* controller.

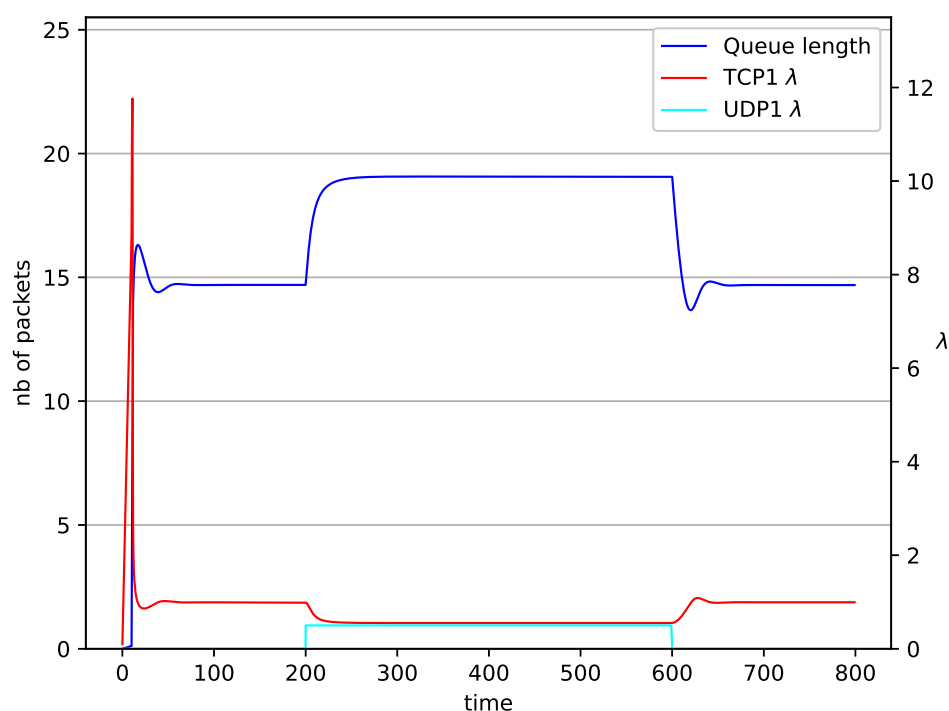


Figure 8. 1 TCP and 1 UDP with RED controller.

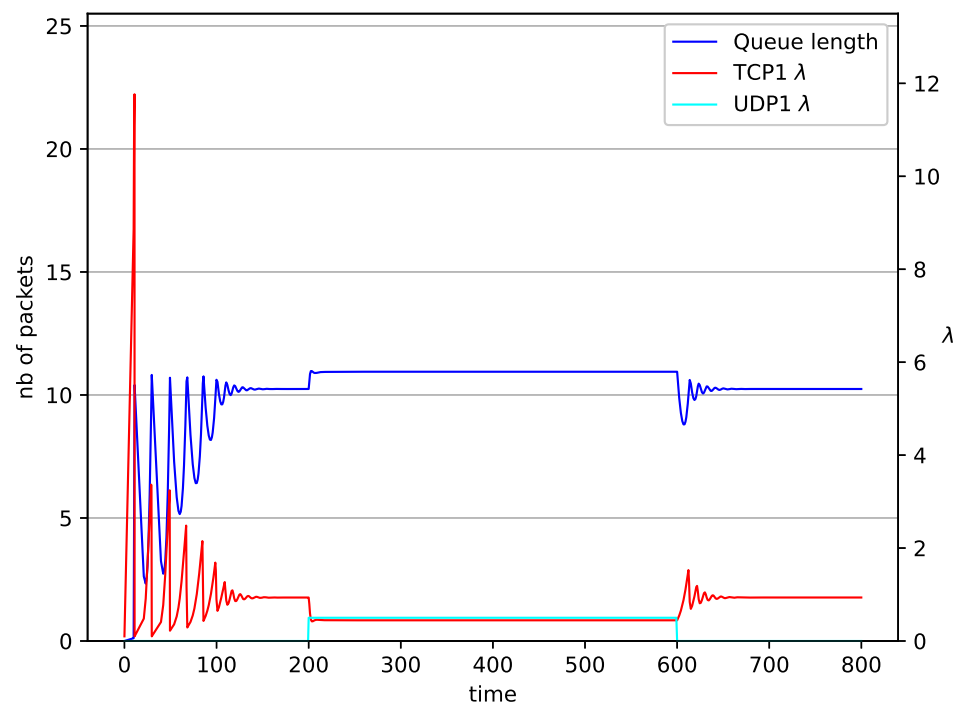


Figure 9. 1 TCP and 1 UDP with NLRED controller.

In Figures 10–13 the case of one TCP and two UDP streams employed is depicted. TCP works all the time. The first constant bit rate UDP stream ($\lambda = 0.5$) starts transmission at time $t = 200$ and finishes at $t = 600$. The second UDP stream ($\lambda = 0.3$) transmits between $t = 350$ and $t = 450$. The second UDP stream causes queue overload. In both cases (PI^γ and RED queue), the queue's occupancy increases dramatically. The PI^γ mechanism slowly stabilizes the queue, reducing the queue size (Figure 11). The RED (Figure 12) and NLRED (Figure 13) mechanisms cannot handle such amounts of input data. In the case of a RED algorithm, the size of the queue oscillates around the point Max_{th} .

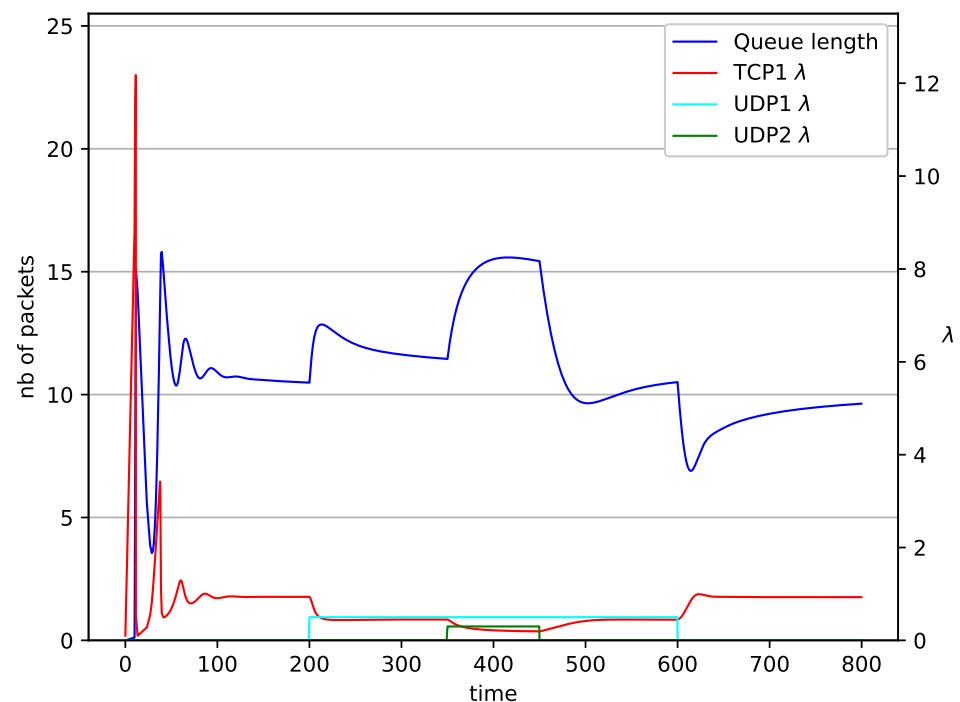


Figure 10. 1 TCP and 2 UDP with PI^γ controller.

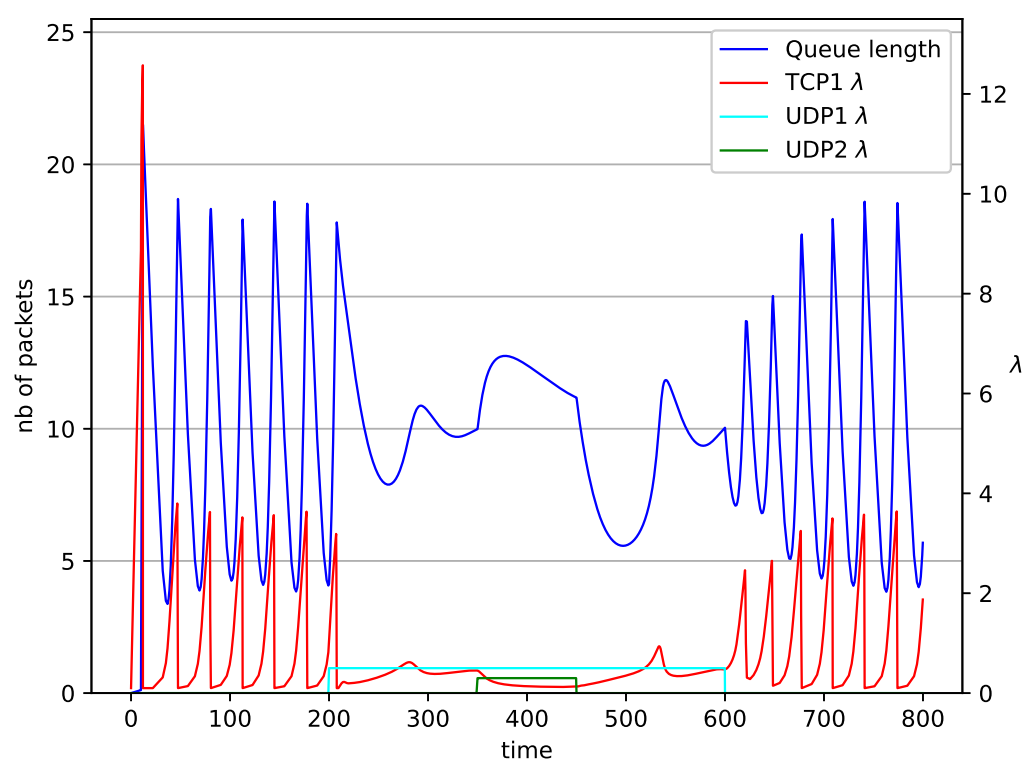


Figure 11. 1 TCP and 2 UDP with *PI* controller.

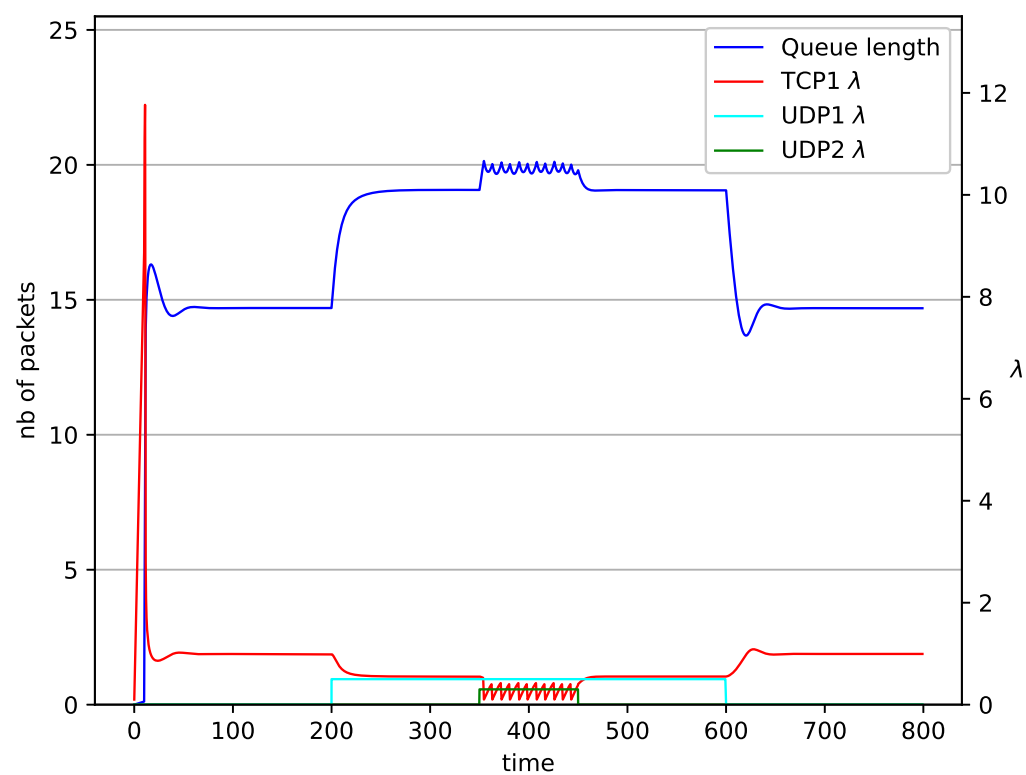


Figure 12. 1 TCP and 2 UDP with RED controller.

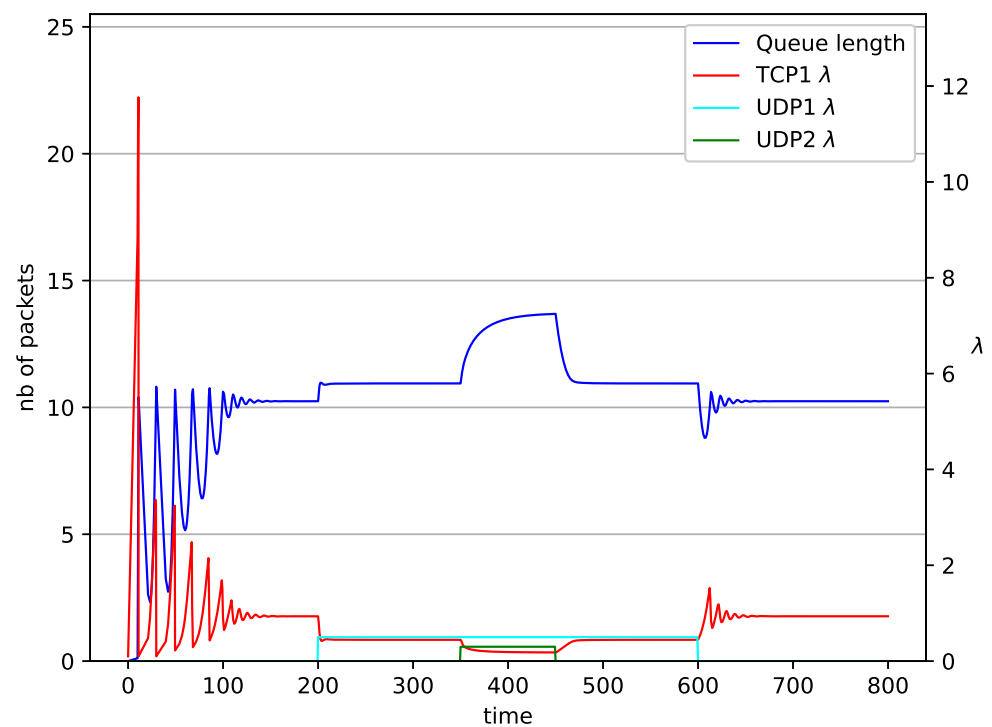


Figure 13. 1 TCP and 2 UDP with NLRED controller.

In Figures 14–17 the situation when two TCP streams cooperate is shown. At the moment when the second stream is switched on, the intensity of both sources stabilises at the same level ($\lambda = 0.5$). In the case of the RED algorithm, the queues occupancies are set at values close to the $Max_{th} = 19.45$. For NLRED mechanism, the average queue size reaches 10.1. In the case of the PI^γ controller, the queue length, after a temporary increase, sags to the desired value. This trend ends when the second source ends its transmission.

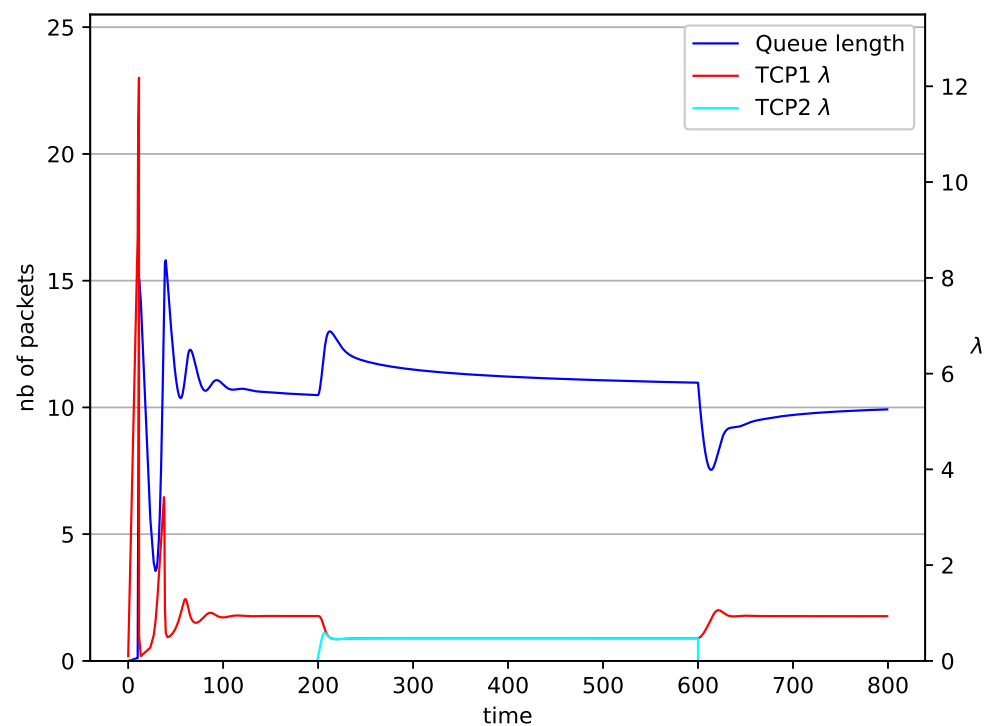


Figure 14. 2 TCP with PI^γ controller.

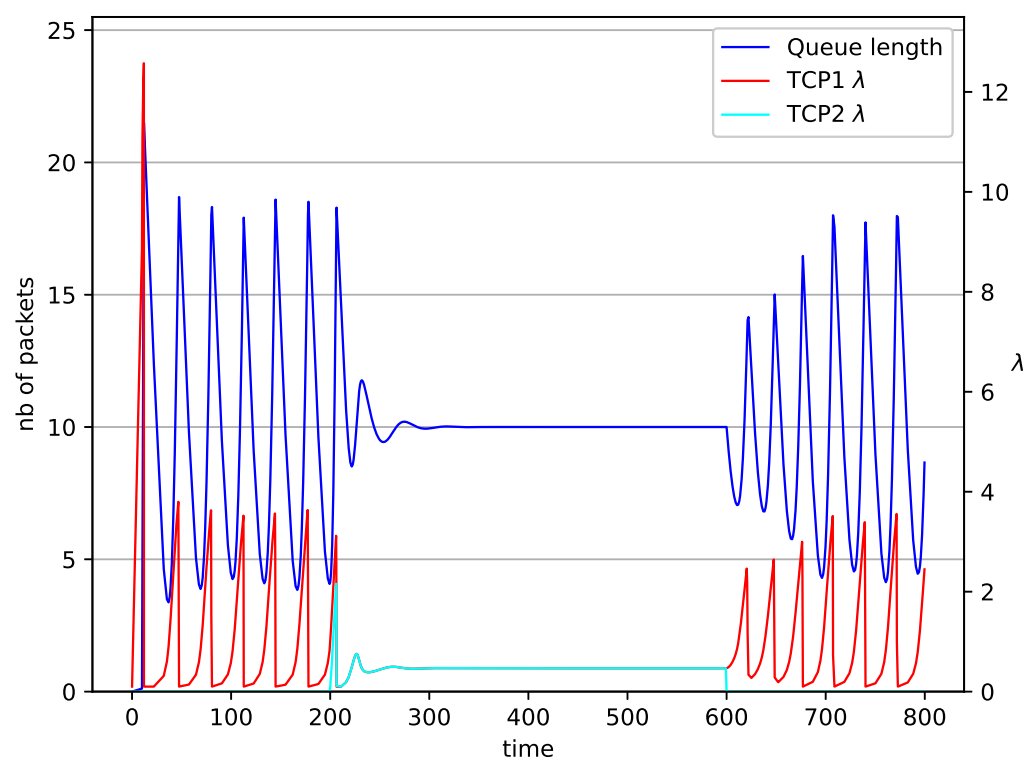


Figure 15. 2 TCP with *PI* controller.

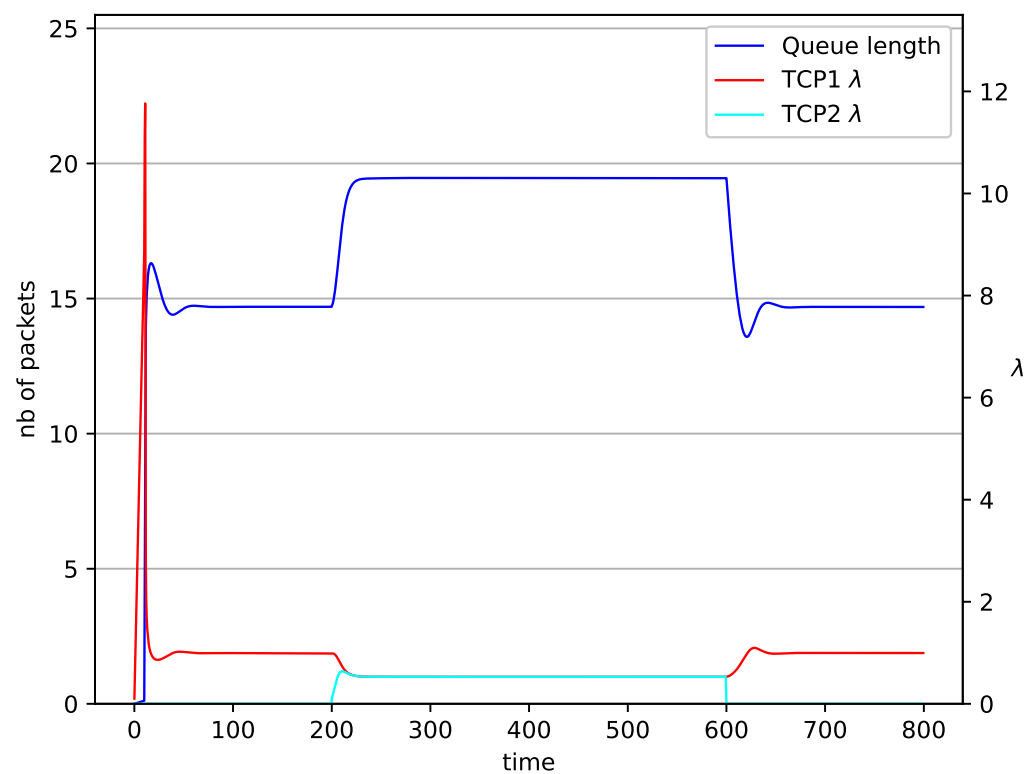


Figure 16. 2 TCP with RED controller.

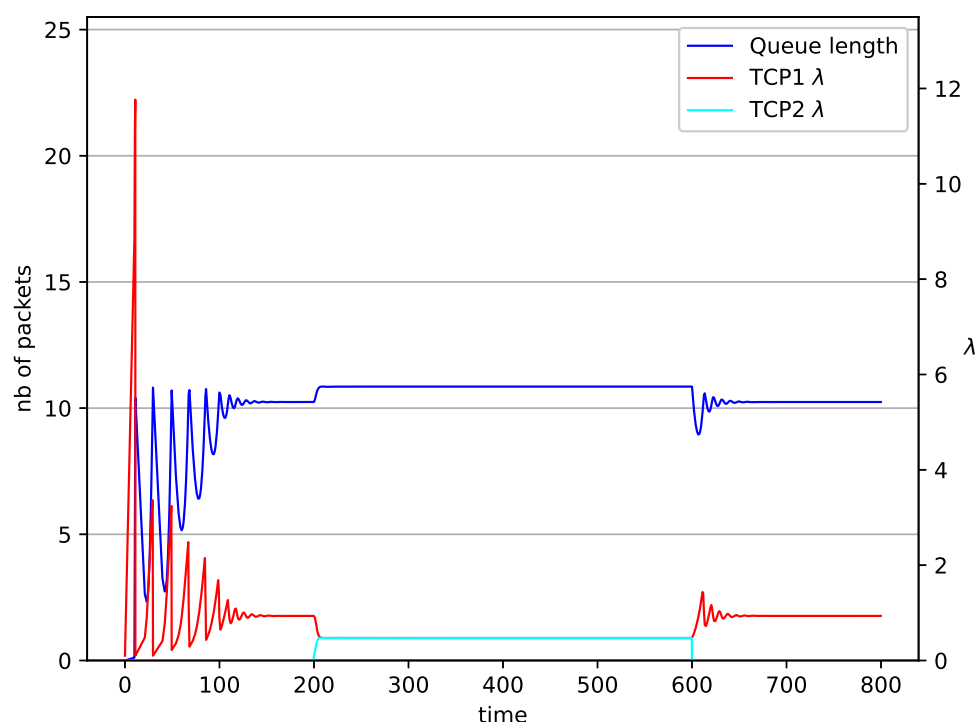


Figure 17. 2 TCP with NLRED controller.

6. Conclusions

This article describes the use of the diffusion approximation to estimate the behavior of the AQM mechanisms, their influence on the evolution of the TCP congestion window and potential cooperation of more TCP/UDP streams. Analytical methods for assessing network mechanisms known from the literature, with the exception of the fluid flow method, are not very suitable for evaluating TCP mechanisms. In the article [17], we have presented a new approach of using the diffusion approximation to model the behavior of the TCP stream. In this article, we expand the given mechanism with the possibility of evaluation of cooperation of more TCP/UDP streams.

The experiments also show the good points of the fractional order controller PI^γ . Decreasing the integration order of the controller reduces the queue fluctuation.

On the other hand, the new approach gets a completely different view on the cooperation of the TCP/UDP protocol with the AQM mechanisms. Our model allows starting and ending the TCP/UDP transmission at any time. The described approach allows us to observe the dynamics of a transmission when different sources start or end their transmission.

The impact of the AQM algorithm on the dynamics of the TCP window is also presented. Our study shows differences in queue behavior in the case of different AQM mechanisms.

Author Contributions: Conceptualization, D.M., A.D., J.D., J.S., T.C. and J.K.; Formal analysis, J.D. and J.K.; Investigation, D.M. and J.S.; Methodology, A.D., D.M. and T.C.; Project administration, J.K.; Software, A.D., D.M. and J.S.; Supervision, A.D. and J.D.; Validation, T.C.; Visualization, D.M. and J.S.; Writing—original draft, A.D. and J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financed by National Science Center project no. 2017/27/B/ST6/00145.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yue, Z.; Zhang, X.; Ren, Y.; Li, J.; Zhong, Q. The performance evaluation and comparison of TCP-based high-speed transport protocols. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering, Beijing, China, 22–24 June 2012; pp. 509–512. [\[CrossRef\]](#)
2. Chydzinski, A. On the Transient Queue with the Dropping Function. *Entropy* **2020**, *22*, 825. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Grazia, C.A.; Patriciello, N.; Klapez, M.; Casoni, M. A cross-comparison between TCP and AQM algorithms: Which is the best couple for congestion control? In Proceedings of the 14th International Conference on Telecommunications (IEEE ConTEL), Zagreb, Croatia, 28–30 June 2017; pp. 75–82.
4. Floyd, S.; Jacobson, V. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.* **1993**, *1*, 397–413. [\[CrossRef\]](#)
5. Feng, C.; Huang, L.; Xu, C.; Chang, Y. Congestion Control Scheme Performance Analysis Based on Nonlinear RED. *IEEE Syst. J.* **2017**, *11*, 2247–2254. [\[CrossRef\]](#)
6. Kahe, G.; Jahangir, A.H. A self-tuning controller for queuing delay regulation in TCP/AQM networks. *Telecommun. Syst.* **2019**, *71*, 215–229. [\[CrossRef\]](#)
7. Feng, W.-C.; Shin, K.G.; Kandlur, D.D.; Saha, D. The BLUE Active Queue Management algorithms. *IEEE/ACM Trans. Netw.* **2002**, *10*, 513–528. [\[CrossRef\]](#)
8. Hu, L.; Kshemkalyani, A.D. HRED: A simple and efficient Active Queue Management algorithm. In Proceedings of the 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969), Chicago, IL, USA, 11–13 October 2004; pp. 387–393.
9. Long, C.; Zhao, B.; Guan, X.; Yang, J. The Yellow Active Queue Management algorithm. *Comput. Netw.* **2005**, *47*, 525–550. [\[CrossRef\]](#)
10. Kelly, F. Mathematical Modelling of the Internet. In *Mathematics Unlimited—2001 and Beyond*; Springer: Berlin/Heisenberg, Germany, 2001; pp. 685–702.
11. Athuraliya, S.; Low, S.H.; Li, V.H.; Yin, Q. REM: Active Queue Management. *IEEE Netw.* **2001**, *15*, 48–53. [\[CrossRef\]](#)
12. Li, Z.; Liu, Y.; Jing, Y. Active Queue Management Algorithm for TCP Networks with Integral Backstepping and Minimax. *Int. J. Control Autom. Syst.* **2019**, *17*, 1059–1066. [\[CrossRef\]](#)
13. Hollot, C.V.; Misra, V.; Towsley, D. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans. Autom. Control* **2002**, *47*, 945–959. [\[CrossRef\]](#)
14. Shah, P.; Yasmin, S.; Asghar, S.; Qayyum, A.; Hasbullah, H. A Fluid Flow Model for SCTP Traffic over the Internet. In Proceedings of the International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 8–9 October 2012; pp. 1–6. [\[CrossRef\]](#)
15. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. Fluid flow approximation of time-limited TCP/UDP/XCP streams. *Bull. Pol. Acad. Sci. Tech. Sci.* **2014**, *62*, 217–225. [\[CrossRef\]](#)
16. Bonaventura, M.; Castro, R. Fluid-flow and packet-level models of data networks unified under a modular/hierarchical framework: Speedups and simplicity, combined. In Proceedings of the 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden, 9–12 December 2018; pp. 3825–3836.
17. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szygula, J.; Marek, D. Diffusion Approximation Model of TCP NewReno Congestion Control Mechanism. *Springer Nat. Comput. Sci.* **2020**, *1*, 43. [\[CrossRef\]](#)
18. Unal, H.; Melchor-Aguilar, D.; Ustebay, D.; Niculescu, S.I.; Ozbay, H. Comparison of PI controllers designed for the delay model of TCP/AQM. *Comput. Commun.* **2013**, *36*, 1225–1234. [\[CrossRef\]](#)
19. Fan, Y.; Ren, F.; Lin, C. Design a PID controller for Active Queue Management. In Proceedings of the Eighth IEEE Symposium on Computers and Communications, ISCC 2003, Kemer-Antalya, Turkey, 3 July 2003; Volume 2, pp. 985–990.
20. Krajewski, W.; Viaro, U. On robust fractional order PI controller for TCP packet flow. In Proceedings of the BOS Conference: Systems and Operational Research, Angers, France, 6–8 March 2014.
21. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. The use of a non-integer order PI controller with an Active Queue Management Mechanism. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 777–789. [\[CrossRef\]](#)
22. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. Self-similarity Traffic and AQM Mechanism Based on Non-integer Order $PI^{\alpha}D^{\beta}$ Controller. *Commun. Comput. Inf. Sci. Springer Int. Publ.* **2017**, *718*, 336–350. [\[CrossRef\]](#)
23. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szygula, J. The Influence of the Traffic Self-similarity on the Choice of the Non-integer Order PI^{α} Controller Parameters. *Commun. Comput. Inf. Sci. Springer Int. Publ.* **2018**, *935*, 76–83. [\[CrossRef\]](#)
24. Szygula, J.; Domański, A.; Domańska, J.; Czachórski, T.; Marek, D.; Klamka, J. AQM Mechanism with Neuron Tuning Parameters. In *Intelligent Information and Database Systems*; Springer: Berlin/Heisenberg, Germany, 2020; pp. 299–311. [\[CrossRef\]](#)
25. Marek, D.; Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Szygula, J. Combined diffusion approximation–simulation model of AQM’s transient behavior. *Elsevier Comput. Commun.* **2020**, *166*, 40–48. [\[CrossRef\]](#)
26. Seferoglu, H.; Modiano, E. TCP-aware backpressure routing and scheduling. In Proceedings of the Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 9–14 February 2014; pp. 1–9. [\[CrossRef\]](#)
27. Bohloulzadeh, A.; Mehri, R. A Survey on Congestion Control Protocols in Wireless Sensor Networks. *Int. J. Wirel. Inf. Netw.* **2020**, *365*–384. [\[CrossRef\]](#)

28. Athanasopoulou, E.; Bui, L.X.; Ji, T.; Srikant, R.; Stolyar, A. Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks. *IEEE/ACM Trans. Netw.* **2013**, *21*, 244–257. [\[CrossRef\]](#)
29. Jan, M.A.; Jan, S.; Alam, M.; Akhunzada, A.; Rahman, I. A Comprehensive Analysis of Congestion Control Protocols in Wireless Sensor Networks. *Mob. Netw. Appl.* **2018**, *23*, 456–468. [\[CrossRef\]](#)
30. Misra, V.; Gong, W.; Towsley, D. Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED. *Comput. Commun. Rev.* **2000**, *30*, 151–160. [\[CrossRef\]](#)
31. Hong, Y.; Yang, O.W.W. Adaptive AQM controllers for IP routers with a heuristic monitor on TCP flows. *Int. J. Commun. Syst.* **2006**, *19*, 17–38. [\[CrossRef\]](#)
32. Sun, J.; Ko, K.; Chen, G.; Chan, S.; Zukerman, M. PD-RED: To improve the performance of RED. *IEEE Commun. Lett.* **2003**, *7*, 406–408.
33. Chen, Y.; Petras, I.; Xue, D. Fractional order control—A tutorial. In Proceedings of the American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 1397–1411.
34. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J.; Szygula, J. The AQM Dropping Packet Probability Function Based on Non-integer Order $PI^\alpha D^\beta$ Controller. In *Lecture Notes in Electrical Engineering*; Springer: Berlin/Heisenberg, Germany, 2019; Volume 496, pp. 36–48. [\[CrossRef\]](#)
35. Czachórski, T.; Pekergin, F. Diffusion Approximation as a Modelling Tool. In *Network Performance Engineering—A Handbook on Convergent Multi-Service Networks and Next Generation Internet, LNCS*; Springer: Berlin/Heisenberg, Germany, 2011; Volume 5233, pp. 447–476. [\[CrossRef\]](#)
36. Czachórski, T.; Grochla, K.; Nycz, T.; Pekergin, F. A diffusion approximation model for wireless networks based on IEEE 802.11 standard. *Comput. Commun.* **2010**, *33*, 86–92. [\[CrossRef\]](#)
37. Nycz, T.; Nycz, M.; Czachórski, T. A Numerical Comparison of Diffusion and Fluid-Flow Approximations Used in Modelling Transient States of TCP/IP Networks. *Commun. Comput. Inf. Sci.* **2014**, *431*, 213–222.
38. Qi, Y.; Ohtsuki, T. QoE-Aware Video Streaming Transmission Optimization Method for Playout Threshold Adjustment in LTE. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5.
39. Zhou, K.; Yeung, K.; Li, V. Nonlinear RED: A simple yet efficient Active Queue Management scheme. *Comput. Netw. Int. J. Comput. Telecommun. Netw.* **2006**, *50*, 3784–3794. [\[CrossRef\]](#)
40. Domańska, J.; Augustyn, D.; Domański, A. The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic. *Bull. Pol. Acad. Sci. Tech. Sci.* **2012**, *60*, 779–786. [\[CrossRef\]](#)
41. Podlubny, I. Fractional order systems and $PI^\lambda D^\mu$ controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [\[CrossRef\]](#)
42. Domański, A.; Domańska, J.; Czachórski, T.; Klamka, J.; Marek, D.; Szygula, J. GPU Accelerated Non-integer Order $PI^\alpha D^\beta$ Controller Used as AQM Mechanism. In *Communications in Computer and Information Science*; Springer: Berlin/Heisenberg, Germany, 2018; Volume 860, pp. 286–299. [\[CrossRef\]](#)
43. Miller, K.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*; Wiley: New York, NY, USA, 1993.
44. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999; Volume 198.
45. Ciesielski, M.; Leszczynski, J. A Numerical Method for Solution of Ordinary Differential Equations of Fractional Order. In *Parallel Processing and Applied Mathematics*; Springer: Berlin/Heisenberg, Germany, 2002; Volume 2328, pp. 695–702. [\[CrossRef\]](#)
46. Newell, G.F. Queues with time-dependent arrival rates. I — The transition through saturation. *J. Appl. Probab.* **1968**, *2*, 436–451. [\[CrossRef\]](#)
47. Gelenbe, E. On Approximate Computer Systems Models. *J. ACM* **1975**, *22*, 261–269. [\[CrossRef\]](#)
48. Gelenbe, E. A Diffusion Model for Packet Travel Time in a Random Multi-hop Medium. *ACM Trans. Sens. Netw. (TOSN)* **2007**, *3*, 10. [\[CrossRef\]](#)
49. Czachórski, T. A method to solve Diffusion Equation with Instantaneous return Processes Acting as Boundary Conditions. *Bull. Pol. Acad. Sci. Tech. Sci.* **1993**, *41*, 417–451.
50. Bonald, T.; May, M.; Bolot, J. Analytic evaluation of RED performance. In Proceedings of the INFOCOM, Tel Aviv, Israel, 26–30 March 2000.
51. Zheng, B.; Atiquzzaman, M. A Framework to Determine the Optimal Weight Parameter of RED in Next-Generation Internet Routers. *Int. J. Commun. Syst.* **2008**, *21*, 987–1008. [\[CrossRef\]](#)