sensors

MDPI

Article

Diffusion Model of Preemptive-Resume Priority Systems and Its Application to Performance Evaluation of SDN Switches

Tomasz Nycz [1], Tadeusz Czachórski [2,*] and Monika Nycz [3]

[1] Department of Distributed Systems and Informatic Devices, Silesian University of Technology, 44-100 Gliwice, Poland; tomasz.nycz@polsl.pl

[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

[3] Department of Computer Networks and Systems, Silesian University of Technology, 44-100 Gliwice, Poland; monika.nycz@polsl.pl

* Correspondence: tadek@iitis.pl

Abstract: The increasing use of Software-Defined Networks brings the need for their performance analysis and detailed analytical and numerical models of them. The primary element of such research is a model of a SDN switch. This model should take into account non-Poisson traffic and general distributions of service times. Because of frequent changes in SDN flows, it should also analyze transient states of the queues. The method of diffusion approximation can meet these requirements. We present here a diffusion approximation of priority queues and apply it to build a more detailed model of SDN switch where packets returned by the central controller have higher priority than other packets.

Keywords: diffusion approximation; transient states; SDN; priority queues

check for updates

Citation: Nycz, T.; Czachórski, T.; Nycz, M. Diffusion Model of Preemptive-Resume Priority Systems and Its Application to Performance Evaluation of SDN Switches. *Sensors* **2021**, *21*, 5042. https://doi.org/10.3390/s21155042

Academic Editor: Joel J. P. C. Rodrigues

Received: 15 June 2021
Accepted: 23 July 2021
Published: 26 July 2021

1. Introduction and an Overview of Existing Results

Software-Defined Networking (SDN), flexible in service of various applications, becomes an alternative to the classical Internet. Traffic and its routing are supervised here by a programmable central controller; its frequent decisions adapt the routing to the current load observed in the network switches, aiming to avoid their congestion. The controller may also activate and deactivate switches to save energy. The network supports multiple classes of traffic having different statistical behavior with different QoS requirements. The service differentiation and QoS provisioning techniques may lead to non-stationarity in the overall traffic of the network. Therefore, traffic intensity in SDN switches is frequently changing. It is recommended that an investigation of performance based on queueing models should allow the transient analysis of packet queues in the switches.

SDN is already an advanced technique. The article [1] presents the history and evolution of programmable networks starting from telephone networks, through packet networks, then Internet, and finally to SDN networks over almost 50 years. An overview of the SDN network, its scalability, elasticity, reliability, and availability are shown in [2]. Reference [3] discusses SDN performance within a data center. Improvements are classified following data, control, and application planes and network type: cloud, wireless and wide-area. The article [4] reviews SND's significant benefits and possible applications. A comparison of various SDN programming languages, such as Flow-based Management Language, then Nettle, Procera, Frenetic, Netcore, Frenetic-OCaml, Pyretic, and NetKAT, is given in [5]. Paper [6] presents an industry survey conducted among IT professionals on network virtualization and SDN within cloud computing, discussing its scalability and roadmap.

Several papers focus on control and data planes issues of SDN controllers; e.g., Reference [7] addresses the problem of logical consistency within data plane nodes when policy rules within data plane nodes are not synchronized with the SDN controller because of network delays, especially within distributed, hierarchical, or flat control planes. The increase in the probability of desynchronization leads to frequent erroneous packet handling or network failure; the article defines requirements that should be met to limit consistency problems. The article [8] examines the validity of SDN dogma, saying that all control should be moved from the data plane to the SDN controller. It may be incorrect for network functions that require only a local view. Middleboxes performing such operations should stay in the data plane to prevent asking a remote SDN controller about local network information; however, their location and rules should be defined in the control plane. The proper selection of controllers within the control plane is reviewed in [9]. Controllers are divided into two groups: centralized (such as beacon, maestro, meridian, or rosemary) and distributed (such as fleet, hyperflow, onix, or smartlight). Additionally, the paper compares the throughput and latencies of selected controllers. The article [10] analyses the usage of SDN mechanisms in the context of wide-area networks; the main focus is on the distribution of SDN controllers within control planes and load balancing, fault tolerance, and monitoring of network nodes within the data plane. The problem of proper placement of SDN controllers is surveyed in [11]. The goal should be achieved by defining the number of required controllers, their location within the network, and their mapping with data-plane nodes based on many network parameters such as latency, resilience, QoS, or general network objectives. Investigated methodologies are divided into two groups, one looking for optimal solutions and the other for heuristic sub-optimal solutions. The problem of proper placement of controllers is also examined in [12]. The authors compare results obtained by different classes of algorithms, such as clustering, integer linear and quadratic programming, evolutionary bio-based, genetic, heuristic, greedy, and simulated annealing based on requirements such as latency, load balancing, fault tolerance, the optimal number of controllers, cost, and control plane communication. In [13], the authors divided existing solutions for controller placement problems into "capacitated" and "uncapacitated" categories. Both categories aim to decrease the number of controller failures; however, the second one does not consider the controller's load and capacity during its placement as a constraint.

Papers concentrating on traffic within SDN networks are surveyed in [14], which also describe strategies that aim to lower the latency within the network. The first strategy involves traffic identification and prediction; the second is based on congestion control; the next one concentrates on load balancing, and the other investigates flow table management. Additionally, edge computing and virtualization are taken into account. The mechanisms of packet forwarding within the SDN network are investigated in [15]. Specifically, forwarding table entries in SDN nodes are described and classified, considering wildcard rules, their priority, validity, placement within multiple tables, and integration of traffic statistics. A review of methods used to predict future traffic and congestion is shown in [16]. The prediction is based on historical and current real-time traffic data. The load balancing and energy-efficient routing within the SDN network are analyzed in [17]. Additionally, solutions for fault-tolerant controller placement problems and end-to-end security challenges are presented. A more detailed review, [18], investigates the problem of green computing concerning SDN networking. The authors present the Green-SDN taxonomy and its detailed analysis and propose a framework to increase its security and efficiency while maintaining reduced energy consumption and environmental impact.

The comparison of SDN architecture with a traditional system is presented in [19]. This article lists the drawbacks of traditional systems and describes SDN networks' management and their further research challenges. The work of [20] analyzes different SDN control plane architectures: centralized, distributed, and hybrid. The comparison is based on numerous factors such as scalability, consistency, reliability, interoperability, controller placement, load balancing, security, etc. Hybrid SDN network architectures are examined in [21]. The approach is a mixture of centralized and decentralized paradigms, e.g., traditional distributed routing algorithms and SDN control plane routing. The article classifies different hybrid SDN models, describes their advantages and disadvantages, and compares them. The article [22] focuses on hybrid SDN architecture; in addition to classifying hybrid models, it presents additional topics such as security and privacy issues, network management with telemetry support, fault tolerance, load balancing, and quality of service. QoS within SDN networks is investigated in [23]; the authors analyze the influence of scalability, consistency, reliability, and load balancing on QoS. Additionally, it examines challenges stemming from interactions between controllers and switches, standards for communication within the control plane and controllers' placement, managing traffic load, and security.

Queueing theory has supported the process of design and performance evaluation of communication systems since the beginning of telephony and telegraphy, that is, since the times of Erlang and Engset. It has great potential to be used in investigations in problems mentioned above and has already been applied in some analyses of SDN performance. The models used until now are mainly elementary queueing systems, i.e., M/M/1 stations with Poisson arrivals and exponential service time distribution, representing switches and controllers, e.g., in [24] Jackson's network model, an open network of M/M/1 queues is used for this purpose. Furthermore, Reference [25] presents a tool for SDN network visualization and performance prediction based on M/M/1 models and shows its results with the use of actual data. In [26], a queueing model is refined by the use M/Geo/1 model where service times are geometrically distributed. A more general model of switch and controller is presented in [27,28]. It is based on a preemption-based packet-scheduling priority scheme where a higher priority is assigned to packets routed to switch from the controller. The solution is approximative; the authors present a method to decompose the system with priority and non-priority queues into two systems with one type of queue. In [29], a Markov model of the switch with preemptive priority and non-priority queues and controller, based on a Markov chain, is presented. The interarrival time distribution is composed of multiple phases, so the model is close to the G/M/1/N station. A similar model with general input and non-preemptive priority was proposed in [30]. A recent article [31] presents a model where the switch is represented by three M/M/n/m stations in series, and the controller is modeled by two similar stations, representing the modules of both devices—switch and controller exchange packets with fixed probabilities. A few other models were based on deterministic or stochastic network calculus [32–34].

All above models have two deficiencies coming from limitations of queueing theory: (i) they are based on the assumption that the network is in a steady state—i.e., the flows are stable, and network metrics such as queueing delays, the length of packet queues in buffers of SDN switches, and packet losses do not depend on time; (ii) they assume exponential interarrival and service time distributions. This is not valid in the case of SDN. QoS-driven routing creates time-dependent traffic and variable network topology, and it is important to understand the behavior of SDN switches affected by sudden changes of paths and routing made by the SDN controller. The flows are not Poisson, and the service times are not exponentially distributed. The sudden changes of flows have performance consequences, including queueing delays and packet losses, which can only be understood via time-dependent transient analysis. However, conventional queueing network theory is poorly adapted to transient analysis. Even in the case of the simplest single-server system, i.e., M/M/1 queue, the transient solution leads to the use of Bessel function expansions; see [35] for infinite and [36,37] for finite M/M/1/N queues. Some particular cases referring to transient queues were analyzed in [38–40]. It is even harder model interconnected systems in transient cases.

Transient behavior of the switch-controller tandem is considered recently in [41]. The authors analyze traffic recorded at a virtual SDN network (mininet). Using a statistical test, they find that it is not stationary. Therefore, they use an approximate transient approach, Pointwise Stationary Fluid Flow Approximation. The balance of input and output flows, taken together with steady-state formulas for M/M/1 and G/M/1 stations, defines the time-dependent evolution of mean queues of switch and controller. It is assumed that the average number of packets at steady state is equal to the average number of packets in non-stationary queue at equilibrium point [42]. Such a model is approximate and limited to mean values (but not distributions) of queues and delays, and it cannot give us, e.g., loss probabilities.

In general, to model stations in a transient regime, the choice of the method includes numerical solution of Markov models, fluid flow approximation, e.g., Reference [43], and diffusion approximation [44]. In Markov models solved numerically, the interarrival and service time distributions may be represented by a system of exponentially distributed phases and fitted to any distribution. Special tools can do this automatically, e.g., Reference [45]. However, this approach is bounded by state explosion; the number of the differential equations (one equation per one state of the model) becomes intractable. Fluid flow approximation, e.g., Reference. [46], similar to the approach presented in [41], may be applied to large topologies. However, it is less exact than the third approach, which is diffusion approximation. We opt for the latter method as it combines transient solutions with the possibilities of including general distributions into the model, and its results are in the form of distributions, not only mean values.

Recently, we have already applied diffusion approximation in modeling a single SDN switch [47] and a network of switches [48]. These models represent an SDN switch as a G/G/1/N station, disregarding communication between the switch and the controller. Here, we develop a diffusion model of a priority station using ideas we proposed in [49], test its quality, and apply it to investigate the communication between the switch and controller. When the flow of an arriving to the switch packet is not identified (it does not exist in the table of flows of the switch), the packet is sent through the uplink channel to the controller to decide on its routing. Then, it returns to the switch with information on its itinerary and is served on a priority basis. Except for the use of the same method, the models and results in [47,48] and here are different. The extension of the presented model into a more complex system of switches and controllers is straightforward.

The rest of the article is organized as follows. Section 2 presents the known diffusion model of a single FIFO station and proposes a new one with priority queues. Section 3 investigates the quality of the priority model using numerical examples, Section 4 presents the rules of a network model composed of single-station models, Section 5 presents an example where diffusion models are implemented to analyze the performance of SDN switch and its communications with the SDN controller, and conclusions are presented in Section 6.

2. Diffusion Single Station Models

2.1. First-In-First-Out G/G/1/N Station

With this method, proposed in [44], the distribution of the number of queued packets in the buffer is represented by the density function of a diffusion process.

The idea comes from the observation that the queue $N(t)$—a discrete stochastic process—and the diffusion proces $X(t)$—a continuous stochastic process—both have normally distributed changes. For any distribution $A(x)$ of interarrival times, with mean $1/\lambda$ and variance σ_A^2, the number of arrivals during an interval Δ tends to the normal distribution with mean $\lambda\Delta$ and variance $\sigma_A^2\lambda^3$. For any distribution $B(x)$ of service times with mean $1/\mu$ and variance σ_B^2, the number of completed services during Δ tends to the normal distribution with mean $\mu\Delta$ and variance $\sigma_B^2\mu^3\Delta$. Therefore, after the interval Δ, the changes in the number of customers present in the queue are subject to the normal distribution with $(\lambda - \mu)\Delta$ and variance $(\sigma_A^2\lambda^3 + \sigma_B^2\mu^3)\Delta$.

The diffusion process with density function, if unrestricted, given by Equation (1)

$$\frac{\partial f(x,t;x_0)}{\partial t} = \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta\frac{\partial f(x,t;x_0)}{\partial x}\,, \tag{1}$$

has normally distributed changes in δt with mean βt and variance αdt; therefore the choice of these parameters

$$\alpha = (\sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3) = C_A^2 \lambda + C_B^2 \mu, \qquad \beta = \lambda - \mu \tag{2}$$

where $C_A^2 = \sigma_A^2 \lambda^2$ and $C_B^2 = \sigma_B^2 \mu^2$ are the square coefficients of variation of $A(x)$, $B(x)$ distributions, enhances similarity of $N(t)$ and $X(t)$.

The diffusion process should be constrained by barriers, following the limitations of a real queue: one barrier is placed at $x = 0$ and the other (if the queue size is limited to N customers) at $x = N$; $X(t) = 0$ means that the queue is empty at time t (idle period of the station), and $X(t) = N$ means that the queue is saturated and the arriving customers are rejected (saturation period). We assume that they correspond to interarrival and service times, but in fact these are rather their residual lifetimes; e.g., the idle time is not the interarrival time but the time between the moments when the last customer in the previous busy period left the system and the first in the next busy period came. Following [44], we assume that the process after a stay at $x = 0$ jumps to $x = 1$ with intensity λ (arrival of a first customer in the new busy period) and jumps from N to $N - 1$ with intensity μ (departure of a customer de-blocking the queue). In this case,

$$\begin{aligned} \frac{\partial f(x,t;x_0)}{\partial t} &= \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta \frac{\partial f(x,t;x_0)}{\partial x} \\ &+ \lambda_0 p_0(t)\delta(x-1) + \mu_N p_N(t)\delta(x-N+1), \end{aligned} \tag{3}$$

$p_0(t)$ and $p_N(t)$ denote the probabilities that the process is at a barrier at time t, and their terms refer to the jumps from barriers. The probabilities of being in the barriers are defined by additional balance equations:

$$\frac{dp_0(t)}{dt} = \lim_{x\to 0}\left[\frac{\alpha}{2}\frac{\partial f(x,t;x_0)}{\partial x} - \beta f(x,t;x_0)\right] - \lambda_0 p_0(t), \tag{4}$$

$$\frac{dp_N(t)}{dt} = -\lim_{x\to N}\left[\frac{\alpha}{2}\frac{\partial f(x,t;x_0)}{\partial x} - \beta f(x,t;x_0)\right] - \mu_N p_N(t). \tag{5}$$

The steady-state solution of the above equations, when the system is in stochastic equilibrium and state probabilities do not depend on time, is given in [44]

$$f(x) = \begin{cases} \dfrac{\lambda p_0}{-\beta}(1 - e^{zx}) & \text{for} \quad 0 < x \le 1\,, \\ \dfrac{\lambda p_0}{-\beta}(e^{-z} - 1)e^{zx} & \text{for} \quad 1 \le x \le N-1\,, \\ \dfrac{\mu p_N}{-\beta}(e^{z(x-N)} - 1) & \text{for} \quad N-1 \le x < N\,, \end{cases} \tag{6}$$

where $z = \frac{2\beta}{\alpha}$. Normalization gives us probabilities p_0 and p_N.

The transient solution of Equations (3)–(5) may be obtained with an analytical-numerical algorithm proposed in [50], used and discussed, e.g., in [51] and recently in [48]. First, the diffusion equation is solved with absorbing barriers at $x = 0$ and $x = N$; i.e., the process is ended when it reaches a barrier. The solution $\phi(x,t;x_0)$ is [52]

$$\phi(x,t;x_0) = \begin{cases} \delta(x - x_0) & \text{for} \quad t = 0\,, \\ \dfrac{1}{\sqrt{2\Pi\alpha t}} \displaystyle\sum_{n=-\infty}^{\infty} \{a(t) + b(t)\} & \text{for} \quad t > 0\,, \end{cases} \tag{7}$$

where:

$$a(t) = \exp\left[\frac{\beta x'_n}{\alpha} - \frac{(x - x_0 - x'_n - \beta t)^2}{2\alpha t}\right],$$
$$b(t) = \exp\left[\frac{\beta x''_n}{\alpha} - \frac{(x - x_0 - x''_n - \beta t)^2}{2\alpha t}\right],$$

and $x'_n = 2nN$, $x''_n = -2x_0 - x'_n$.

Then, the density of the diffusion process having barriers with jumps is expressed with the use of functions $\phi(x,t;x_0)$

$$f(x,t;\psi) = \phi(x,t;\psi) + \int_0^t g_1(\tau)\phi(x,t-\tau;1)d\tau + \int_0^t g_{N-1}(\tau)\phi(x,t-\tau;N-1)d\tau\ . \quad (8)$$

where $g_1(t)$ and $g_N(t)$ are derived with the use of balance Equations (4) and (5).

This is the transient solution, but it assumes constant parameters of equations. If they are changing with time, e.g., if the flow intensity λ is time-dependent, and, in consequence, we have $\alpha(t)$ and $\beta(t)$, the diffusion equation is solved in short time intervals where the parameters of the equation are considered constant and change their values only with the change in the interval. The solution at the end of an interval is used as the initial condition for the next interval.

The solution $f(x,t)$ approximates the distribution of the queue length. The density of the queue latency (response time) is obtained with the use of the first passage time; i.e., the time the process needs to walk a certain distance. The density function $\gamma_{x_0,0}(t)$ of the first passage time from $x = x_0$ to $x = 0$,

$$\begin{aligned}\gamma_{x_0,0}(t) &= \frac{\partial}{\partial t}\int_{0+}^{\infty}\phi(s,t;x_0)dx = lim_{x\to 0}[\frac{\alpha}{2}\frac{\partial}{\partial x}\phi(x,t;x_0) - \beta\phi(x,t;x_0)] \\ &= \frac{x_0}{\sqrt{2\Pi\alpha t^3}}e^{-\frac{(x_0+\beta t)^2}{2\alpha t}}. \end{aligned} \quad (9)$$

A new customer who joins the queue at time t has, with probability density $f(x,t)$, x customers ahead him. The queueing delay is equivalent to the time the process needs to go from the initial point x to 0 (corresponding to the customer service). The pdf of the delay introduced by the queue length distribution with density $f(\xi,t;\psi)$ is then

$$f_R(x,t) = \int_0^N \gamma_{\xi,0}(x)f(\xi,t;\psi)d\xi. \quad (10)$$

The input traffic may be non-homogeneous, composed of independent flows called *classes*, $k = 0,1,\ldots K$ that have input parameters $\lambda^{(k)}$, ${\sigma_A^{(k)}}^2$ specific to each class and service parameters $\mu^{(k)}$, ${\sigma_B^{(k)}}^2$ waiting for service in the common FIFO queue. In this case, the number of all class customers coming to the system has a normal distribution with mean and variance being the sum of corresponding means and variances. The input and service parameters for the total flow of customers are [53]

$$\lambda = \sum_{k=0}^{L}\lambda^{(k)}, \quad C_A^2 = \sum_{k=1}^{L}\frac{\lambda^{(k)}}{\lambda}{C_A^{(k)}}^2\ , \quad (11)$$

$$\frac{1}{\mu} = \sum_{k=1}^{L}\frac{\lambda^{(k)}}{\lambda}\frac{1}{\mu^{(k)}}\ , \quad C_B^2 = \mu^2\sum_{k=1}^{L}\frac{\lambda^{(k)}}{\lambda}\frac{1}{{\mu^{(k)}}^2}({C_B^{(k)}}^2 + 1) - 1\ , \quad (12)$$

where $\lambda^{(k)}/\lambda$ is the probability that a customer belongs to a class k. The diffusion process where α and β have the above parameters gives the approximation of $p(n)$ and the distribution of the total number of customers in the station, and then for any class k, the distribution $p^{(k)}(\nu)$

$$p^{(k)}(\nu) = \sum_{n=\nu}^{N} [p(n)\binom{n}{\nu}(\frac{\lambda^{(k)}}{\lambda})^{\nu}(1 - \frac{\lambda^{(k)}}{\lambda})^{n-\nu}] \,, \qquad k = 0, \ldots, L \,. \tag{13}$$

2.2. Preemptive-Resume G/G/1/N/PRIOR Station

The above classic model of G/G/1/N station with FIFO queue may be extended to the case of multiple classes of customers, with each class having its own priority. Depending on the type of priorities, the service of these clients is different. There are three categories of interrupted service queues: (1) postponable, (2) preemptive-resume, and (3) preemptive repeat. The first category assumes that when a new client with higher priority comes to the system, he waits for the end of the service of the currently serviced client (the current service is not interrupted). The second and third category assume interruption with the service of the currently serviced client and the start of the service of the new client. However, after the end of the service of the higher privileged clients, the preemptive-resume interruption continues the service of the client and preemptive-repeat starts the client service from the beginning. A good review of classical models of priority systems is given in [54]. It refers in general to M/G/1/PRIOR steady-state models. Below, we deal with preemptive-resume priorities. Our model, similarly to in the case of one class of customers, assumes general distributions of interarrival times and service times at each priority level and limited to N number of customers of each priority.

We keep the notation described in the previous section of adding upper index (k) to identify the priority class $k = 0, 1, \ldots L$, $k = 0$ as the highest priority, and $k = L$ as the lowest. This way, $1/\lambda^{(k)}$ and $\sigma_A^{(k)^2}$ refer to the mean and variance of interarrival times of class k customers, and $1/\mu^{(k)}$ and $\sigma_B^{(k)^2}$ refer to the mean and variance of their service times; $p_0^{(k-1)}(t)$ is probability that at time t there are no customers of class k in the system.

We will also consider a diffusion process $X^K(t)$, which refers to the joint number of customers of classes $0 \ldots K$ in the system; parameters α^K and β^K refer to its movement, and $f^K(x, t; x_0)$ denotes its pdf. With the same arguments as for one class in the previous section, we may say that the number of customers of several classes counted jointly at arrival and departure has a normal distribution. The diffusion process may describe the evolution of this number of customers in the system. However, only the input processes of these classes are independent. The output process of a class k is dependent on the processes of all higher classes: the service of a customer of class k may be finished only if customers of classes $0 \ldots k-1$ are absent in the system. Therefore, the parameters α^K, β^K may be written as

$$\begin{aligned} \alpha^K &= \sum_{k=1}^{K} \lambda^{(k)} C_A^{(k)^2} + \sum_{k=1}^{K} \Big((1 - p_0^{(k-1)}(t)) \mu^{(k-1)} C_B^{(k-1)^2} \Big) \\ &\quad + \sum_{k=1}^{K} \Big(p_0^{(k-1)}(t) \mu^{(k)} C_B^{(k)^2} \Big), \\ \beta^K &= \sum_{k=1}^{K} \lambda^{(k)} - \sum_{k=1}^{K} \Big((1 - p_0^{(k-1)}(t)) \mu^{(k-1)} \Big) - \sum_{k=1}^{K} \Big(p_0^{(k-1)}(t) \mu^{(k)} \Big), \end{aligned} \tag{14}$$

where

$$C_A^{(k)^2} = \sigma_A^{(k)^2} \lambda^{(k)2}, \quad C_B^{(k)^2} = \sigma_B^{(k)^2} \mu^{(k)2}.$$

Let $v^{(K)}(n,t)$ denote the probability that n customers of class K are present at time t in the system, and let $p^{K-1}(n,t)$ denote the probability that n customers of all classes $0,\dots K-1$ are present at time t in the system. Obviously, for the highest priority class

$$v^{(0)}(n,t) = p^{0}(n,t)$$

and for other classes

$$p^{K}(n,t) = \sum_{\nu=0}^{n} p^{K-1}(n-\nu,t)v^{(K)}(\nu,t), \quad K = 1,\dots,L, \quad n = 0,1,\dots,N$$

or

$$v^{(K)}(n,t) = \frac{p^{K}(n) - \sum\limits_{\nu=0}^{n-1} p^{K-1}(n-\nu,t)v^{(K)}(\nu,t)}{p^{K-1}(0,t)}, \quad K = 1,\dots,L, \quad n = 0,1,\dots,N \tag{15}$$

Note that index K refers to classes $0,\dots,K$ and index (K) to the single class K.

Denote by $E[n^{(k)}(t)]$ the mean number of customers class k present in the system

$$E[n^{(k)}(t)] = \sum_{\nu=0}^{\infty} v^{(k)}(\nu,t)\nu$$

and by $E[n^{K}(t)]$ the mean number of customers class $0\dots K$ in the system

$$E[n^{K}(t)] = \sum_{n=0}^{\infty} p^{K}(n,t)n,$$

A sketch of the algorithm is as follows:

- $K = 0$: we consider the highest priority class $k = 0$ alone and use the single class model presented in the previous section. The customers of lower classes are transparent for $k = 0$ class; therefore, the solution is correct. $f^{(0)}(n,t;x_0) = p^{(0)}(n,t) = v^{(0)}(t)$.
- $K = 1$: we consider two classes, $k = 0,1$, determine $\alpha^{(1)}$, $\beta^{(1)}$ following (14), solve the diffusion equation with these parameters to obtain $f^{(1)}(x,t;x_0)$, which approximates the distribution $p^{1}(n,t)$ of the joint number of customers of classes $k = 0$ and $k = 1$; we then compute $v^{(1)}(n,t)$.
- $K = 2$: we consider the system with three classes, $k = 0,1,2$ to determine the parameters $\alpha^{(2)}$, $\beta^{(2)}$ following (14), solve the diffusion equation to obtain $f^{2}(x,t;x_0)$ and $p^{2}(n,t)$, then, using $p^{(1)}(n,t)$ of the previous step , compute $v^{(2)}(n,t)$, etc., until $K = L$.

Note that for the mean values, $E[n^{K}(t)] = E[n^{K-1}(t)] + E[n^{(K)}(t)]$.

Before analyzing the waiting times, we have to define the distribution of the completion time. The completion time is the period between the start and the end of any customer service. On the highest priority level, the completion time is equal to the service time; for other classes, it also includes interruptions caused by the arrival and service of higher-priority customers. Suppose T is the service time of a customer of class k. If n customers of classes $1,\dots,k-1$ arrive during the time T, the service will be interrupted n times; n has an approximately normal distribution with the mean $\sum_{l=0}^{k-1}\lambda^{(l)}T$ and the variance $\sum_{l=0}^{k-1}\lambda^{(l)}C_A^{(l)^2}T$.

The duration of any of n breaks is distributed like the busy period $\gamma^{(k-1)}$ of the system serving customers of classes $0,\dots,k-1$. The busy period starts with the arrival of a customer to the empty system and lasts until the moment when the system becomes empty. Its duration may be seen as the first passage time from $x_0 = 1$ (first customer arrives) to $x = 0$ (nobody in the system) and is given by Equation (9) with parameters corresponding to the diffusion process with $K-1$ classes. For the sake of simplicity, we neglect here the weak probability that the process, before it comes to zero, may reach the upper barrier at N, stay there, jump to $N-1$, come back to N, etc.

The total time of breaks in T has the pdf

$$\varphi^{(k)}(t \mid T) = \sum_{n=0}^{\infty} p_{n|T} \gamma^{(k-1)(*n)}(x)$$

where $p_{n|T}$ is the probability of n breaks in T and $\gamma^{(k-1)(*n)}(t)$ is the n-fold convolution of $\gamma^{(k-1)}(t)$ with itself. Thus the pdf $c^{(k)}(t)$ of the completion time is

$$c^{(k)}(t) = \int_0^{\infty} b^{(k)}(t) \varphi^{(k)}(t-T \mid T) \mathbf{1}(t-T) dT,$$

where $\mathbf{1}(t) = 0$ for $t < 0$ and $\mathbf{1}(t) = 1$ for $t \geq 0$, and from its Laplace transform

$$c^{(k)}(s) = \int_0^{\infty} b^{(k)}(T) e^{-sT} \sum_{n=0}^{\infty} \{p_{n/T} [\bar{\gamma}^{(k)}(s)]^n\} dT$$

we obtain its moments $E = [c^{(k)}]$ and $E[(c^{(k)})^2]$:

$$\begin{aligned}
E = [c^{(k)}] &= -\frac{d}{ds} c^{(k)}(s)_{s=0} = \{E[\gamma^{(k-1)}] \Lambda^{(k-1)} + 1\} \frac{1}{\mu^{(k)}}, \\
E[(c^{(k)})^2] &= \frac{d^2}{ds^2} c^{(k)}(s)_{s=0} = E[\gamma^{(k-1)}]^2 \Big[\Big(\sum_{l=0}^{(k-1)} \lambda^{(l)} C_A^{(l)^2} \Big) \frac{1}{\mu^{(k)}} \\
& \quad -\Lambda^{(k-1)} \frac{1}{\mu^{(k)}} \Big] + E[(\gamma^{(k-1)})^2] \Lambda^{(k-1)} \frac{1}{\mu^{(k)}} + \\
&+ \quad E[\gamma^{(k-1)}] E[(b^{(k)})^2] \Lambda^{(k)} \cdot \\
& \quad \cdot \{E[\gamma^{(k-1)}] \Lambda^{(k-1)} + 2\} + E[(b^{(k)})^2].
\end{aligned}$$

where $\Lambda^{(k)} = \sum_{l=0}^{k} \lambda^{(l)}$.

When all input streams are Poisson, i.e., $C_A^{(l)^2} = 1, l = 1, \ldots, k$, the results are identical to the exact formulae given for this case in [54].

Finally, similarly as in Equation (10), we can define the pdf of the delay (response time) at every priority level k:

$$f_{R^{(k)}}(x,t) = \int_0^N \gamma_{\xi,0}^{(k)}(x) f^{(k)}(\xi, t; \psi) d\xi. \tag{16}$$

In the pdf of the first passage time $\gamma_{\xi,0}^{(k)}(x)$ for a priority k, the mean and variance of the service time should be replaced by the mean and variance of the completion time $c^{(k)}$. Mean waiting time $E[w^{(k)}]$ to start the service is

$$E[w^{(k)}] = E[R^{(k)}] - E[c^{(k)}].$$

3. Validation of the Priority Model

Diffusion approximation remains a heuristic approach, and we do not know strict bounds on its errors; therefore, we should check its quality in various cases. The errors of the method in case of FIFO queue as presented in Section 2.1 were discussed, e.g., in [46,51]. Below, we investigate the quality of the diffusion priority model, considering a few numerical examples differing in the number of priorities, input intensities, and type of interarrival and service time distributions.

The first three cases concentrate on various input intensities and two priority classes. The fourth case investigates a system with three priorities. The fifth and final scenario is dedicated to non-exponential interarrival distribution. In all cases, the results of the diffusion model have been validated by comparison with the ones obtained with OMNeT++ discrete network simulator [55]. The standard OMNET++ package can only collect steady-state results; therefore, we adapted it to the needs of transient analysis by modifying packet generators, algorithms of collecting statistics, and handling message mechanisms. The simulation results are averaged over 100,000 independent runs.

3.1. Two Priorities, Low Input Intensities

This scenario considers two classes of customers. The arrival rate of the priority class changes in the following way: $\lambda^{(0)} = 0.4$ during intervals $t \in [0,10], [20,30], [40,50], \ldots$, and $\lambda^{(0)} = 0$ between these intervals. Time is expressed in generic units. The non-priority customers arrive with constant intensity $\lambda^{(1)} = 0.4$. Both queues are limited to $N^{(0)} = N^{(1)} = 20$; i.e., the system can host up to 20 customers of each class but no more than 20 in total. When active, the input streams are Poisson; exponential service time distributions for both classes are the same: $\mu^{(0)} = \mu^{(1)} = 1$. The system is stable because its maximum utilization factor is $\rho = 0.8$, but the considered intervals are too short to allow it to attain a steady-state.

The total number of customers of both classes displayed in Figure 1 increases when the priority customers come into the system. In the remaining periods of time, the service ($\mu = 1$) is fast enough to countermeasure the non-priority intensity ($\lambda = 0.4$) and makes the queue effectively decrease until new priority clients arrive at the system.

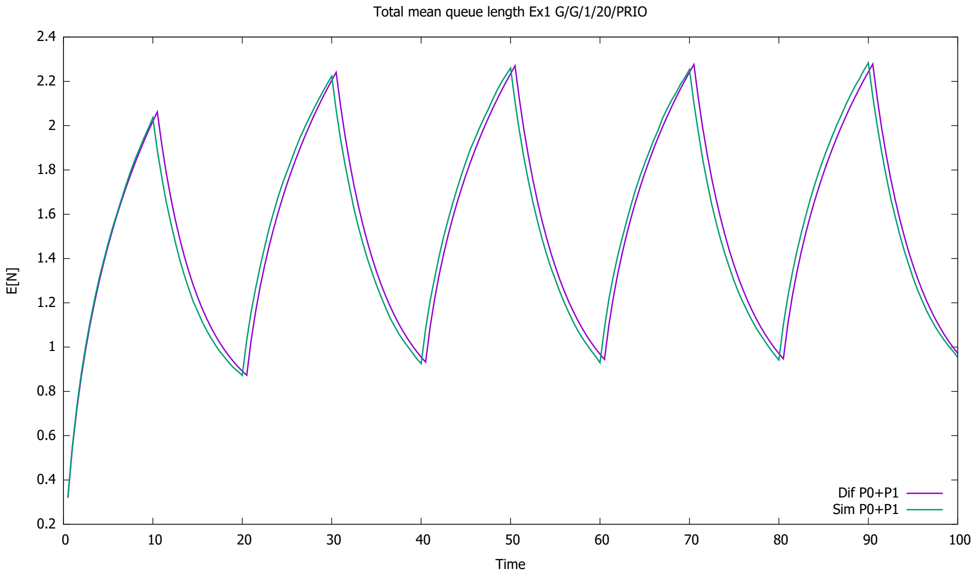

Figure 1. Section 3.1, low load: total mean queue length as a function of time for both classes (P0 + P1) taken together.

Figure 2 displays the mean number of customers of each class as a function of time. The diffusion and simulation results are compared. We see how when both flows are active, mainly priority customers are served, and the non-priority clients are queued and wait their turn—their queue increases almost linearly. For every 10 time units in which priority customers do not come, they have more chance to enter the service, and their queue empties.

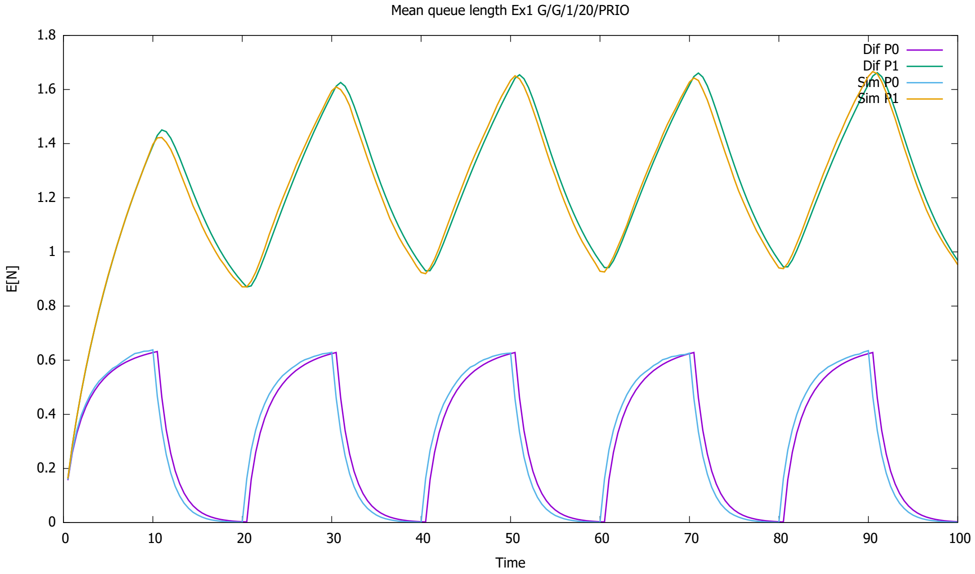

Figure 2. Section 3.1, low load: mean queue lengths as a function of time for priority (P0) and non-priority (P1) classes, and diffusion and simulation results.

Figure 3 shows probabilities $p_0^{(0)}(t)$ and $p_0^{(1)}(t)$ of empty queue for both classes. The probabilities decrease during higher traffic periods and increase elsewhere; we see the influence of the priority traffic on non-priority traffic, with a constant arrival rate.

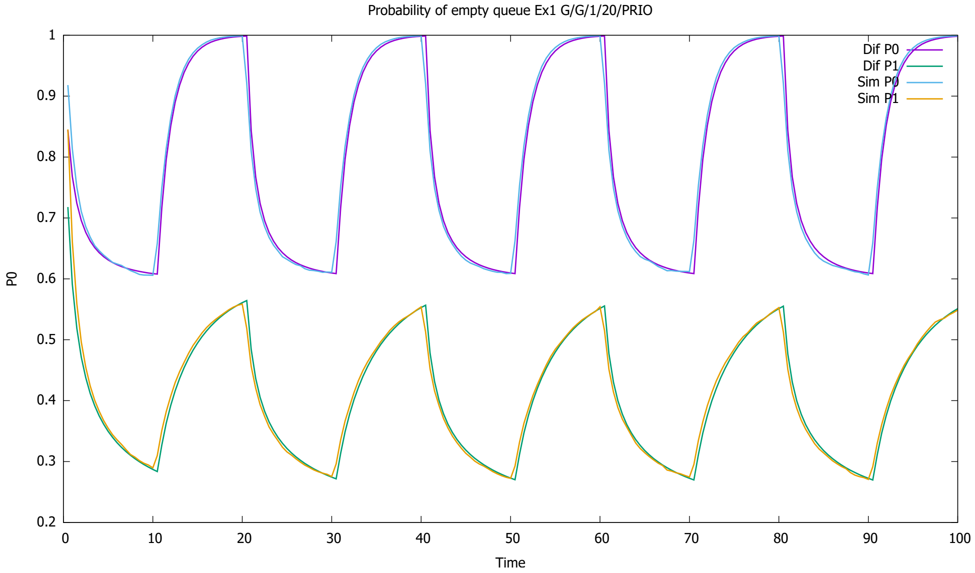

Figure 3. Section 3.1, low load: probabilities of empty queues for priority (P0) and non-priority (P1) classes.

Similarly, probabilities $p_N^{(0)}(t)$ and $p_N^{(1)}(t)$ are displayed in Figure 4. As the probability of buffer overflow is weak, we observe that the series of 100,000 simulation runs is not sufficient to determine it properly. No overflow is observed in simulation for the priority class, and the simulation results for the non-priority class are distorted. There is no numerical problems in case of diffusion approximation, even if results are in the order of 10^{-18}.

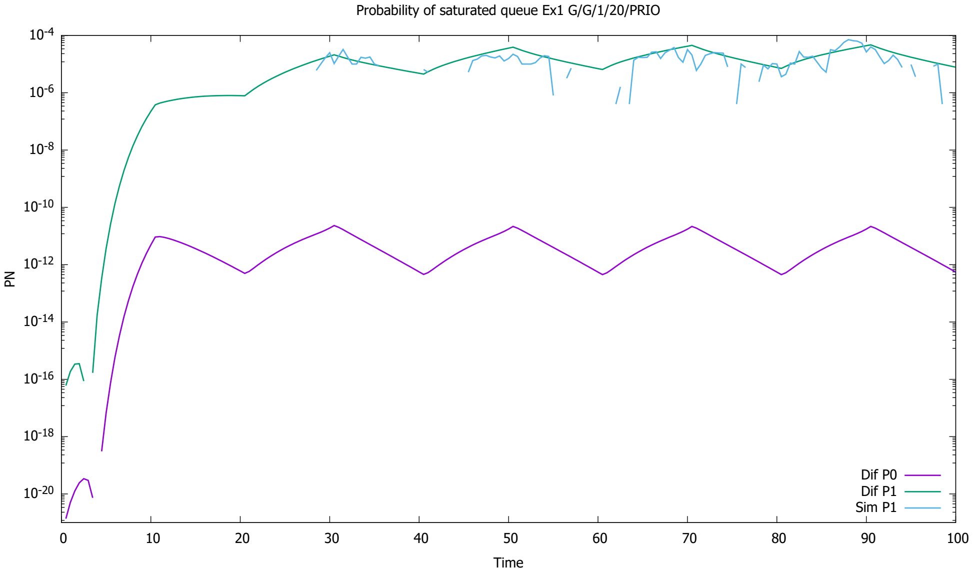

Figure 4. Section 3.1, low load: probabilities of saturated queues as a function of time for priority (P0) and non-priority (P1) classes.

3.2. Two Priorities, Medium Input Intensities

We keep the same pattern of the traffic but increase its intensity. The rate of priority traffic is ($\lambda^{(0)} = 1.2$), i.e., three times higher than previously in Section 3.1 in the same intervals $t \in [0, 20]$, $[40, 60]$, $[80, 100]$ and zero otherwise. The intensity of the non-privileged class is higher by 0.1 ($\lambda^{(1)} = 0.5$), and constant.

The system is unstable during active periods of priority traffic; the service station is not able to serve all incoming priority customers. During every first 20 time-units in 40-unit cycles, the first-class customers are serviced and queued, unlike second-class customers, who are only queued. For the next seven time units, on average, the class 1 service continues, which results in further queuing of class 2. During the last 13 units of the cycle, on average, the service of priority clients ends, and the non-priority begins. However, the low traffic period is too short to allow the service of all accumulated non-priority customers. The lower priority queue is gradually increasing cycle by cycle, and the same is true for the total number of customers of both classes; see Figures 5 and 6.

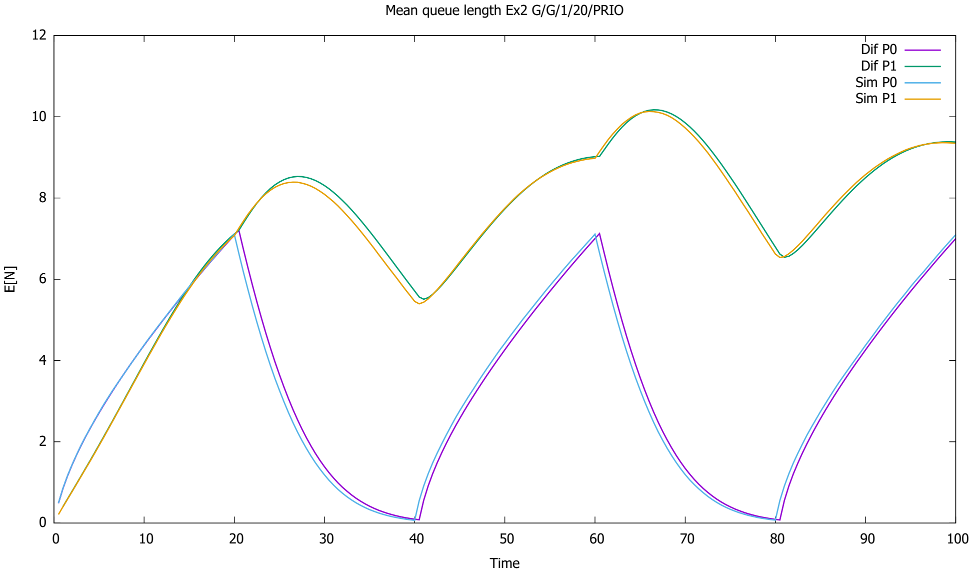

Figure 5. Section 3.2, medium load: mean queue lengths of priority (P0) and non-priority (P1) classes.

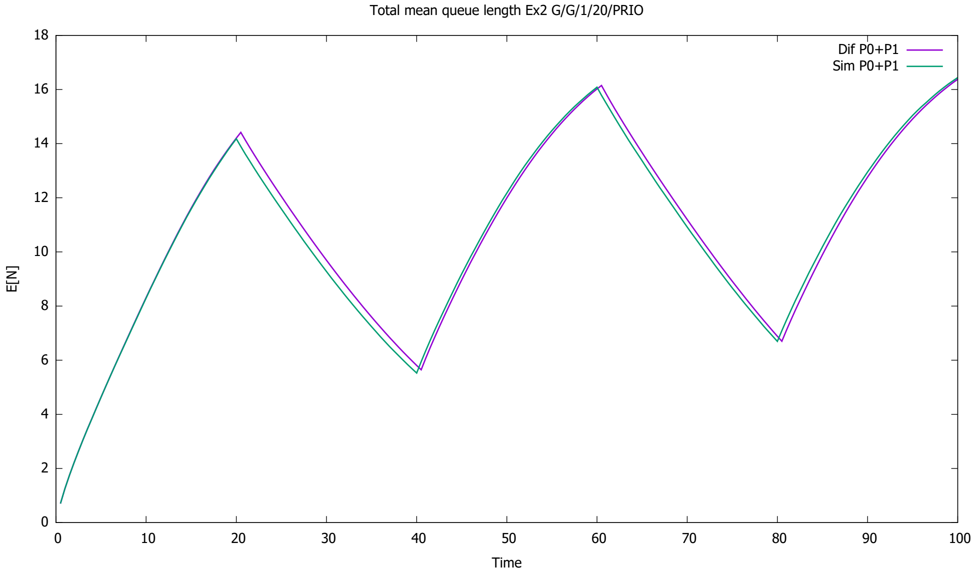

Figure 6. Section 3.2, medium load: total mean queue length for both classes together.

The changes in the system highly affect the probability of both empty (Figure 7) and full (Figure 8) queues of classes 1 and 2. Again, the observation starts with an empty queue, but this time the probability of an empty queue for non-priority drops to zero. Moreover, the probability that the saturated non-priority queue increases with each cycle is grows a little bigger, signaling that the system will saturate in the future.

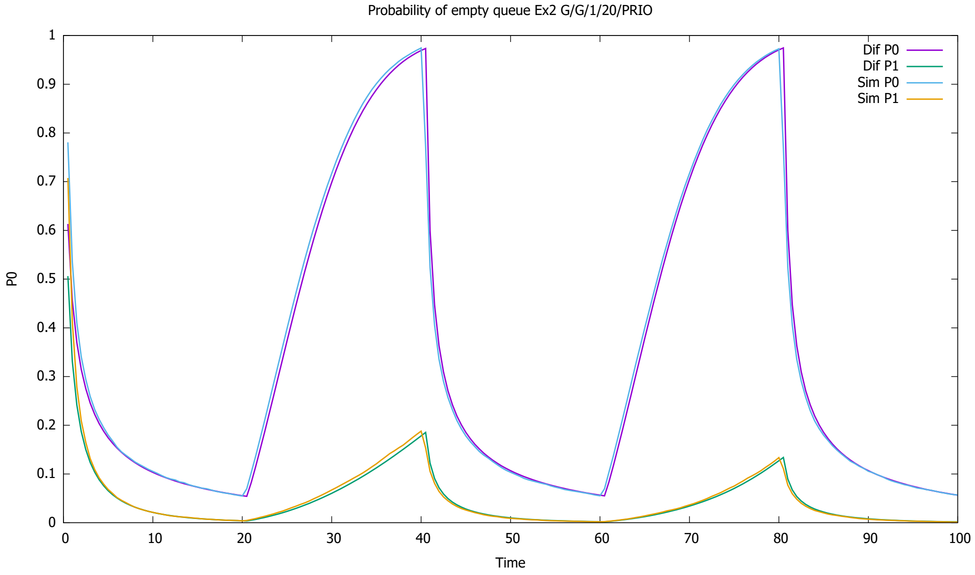

Figure 7. Section 3.2, medium load: probabilities of empty queue $p_0(t)$ for priority (P0) and non-priority (P1) classes.

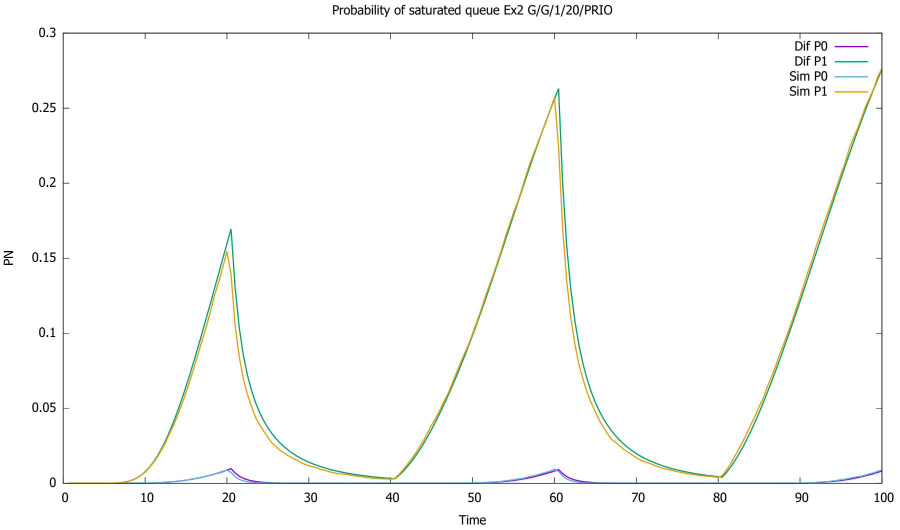

Figure 8. Section 3.2, medium load: probabilities of saturated queues $p_N(t)$ for priority (P0) and non-priority (P1) classes.

3.3. Two Priorities, High Input Intensities

The input intensity is two times higher ($\lambda^{(0)} = 2.4$) than in the previous Section 3.2. The lower class intensity remains constant and is twice as high previously, ($\lambda^{(1)} = 1$). The intervals and other assumptions remain the same as in Section 3.2.

Initially, both queues increase, as shown in Figure 9. After a while, the non-priority queue drops almost to zero because the buffer is monopolized by the higher class. Only when priority customers cease to arrive do lower-priority clients have a chance to enter the buffer, and their queue increases. Therefor the cycles of queue changes are interleaved: when the priority queue increases, the non-priority queue decreases, and vice-versa. There is not space enough in the buffer for both classes; Figure 10 shows that for most of the time, it is full or almost full. The system remains stable due to massive losses; see probabilities of the empty queue in Figures 11 and 12. The probability that the priority queue is full reaches its maximum at the ends of the customer arrival cycles and then decreases. Only at these moments do the lower-class clients see that the queue may be available.

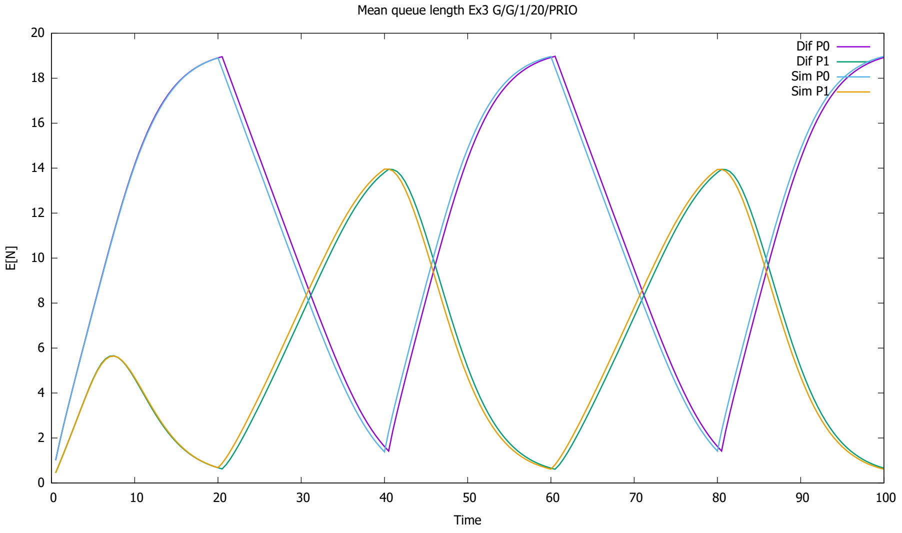

Figure 9. Section 3.3, high load: mean queue lengths of priority (P0) and non-priority (P1) classes.

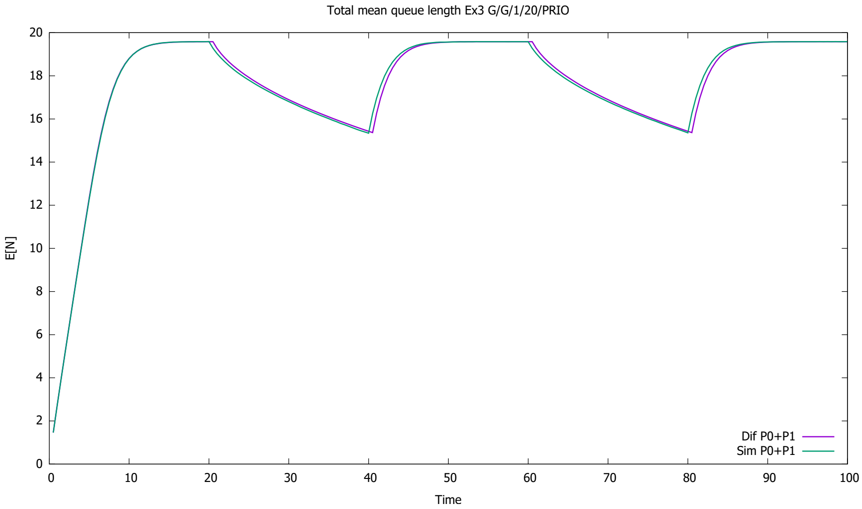

Figure 10. Section 3.3, high load: total mean queue length of both priority classes.

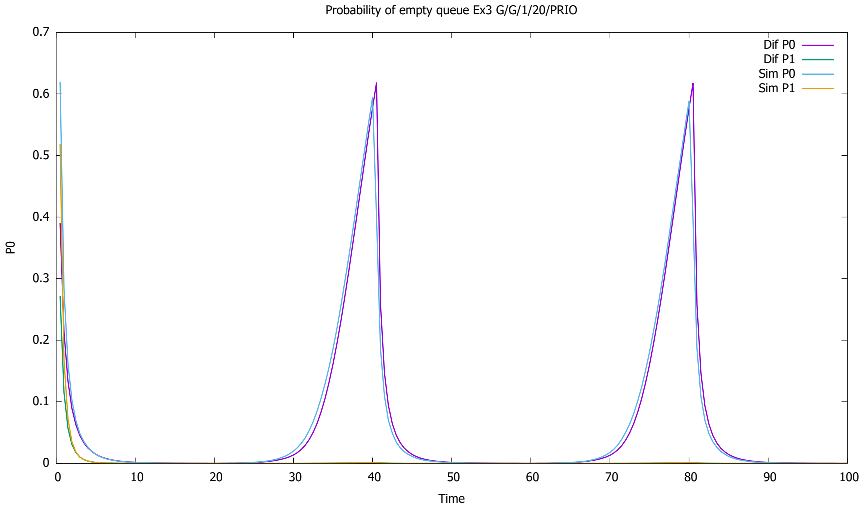

Figure 11. Section 3.3, high load: probabilities of empty queues for priority (P0) and non-priority (P1) classes.

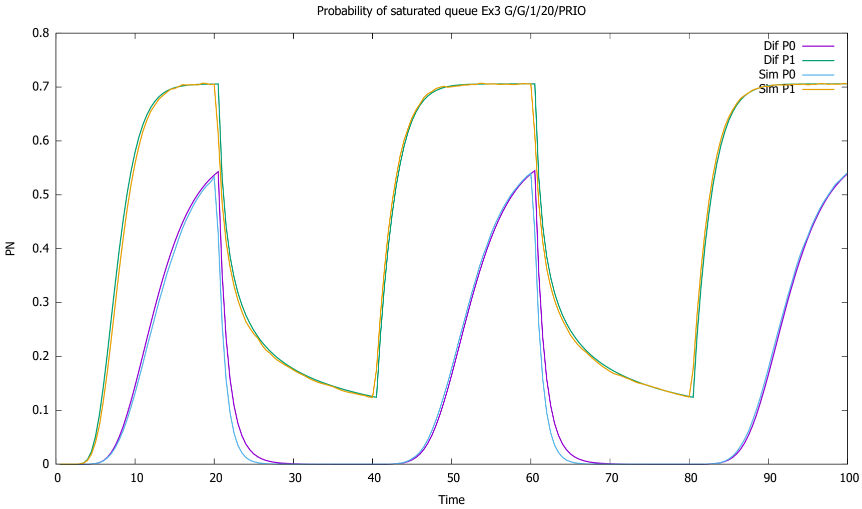

Figure 12. Section 3.3, high load: probabilities of saturated queues for priority (P0) and non-priority (P1) classes.

3.4. Three Priorities; Mean Service Time Depends on Priority

This time, the server is designed to handle three priority levels. The highest-priority customers come with intensity $\lambda^{(0)} = 0.25$ during the same intervals as in Sections 3.2 and 3.3 and otherwise $\lambda^{(0)} = 0$. The intensity of medium-priority customers is constant, $\lambda^{(1)} = 0.5$, while the lowest-priority customers have constant intensity $\lambda^{(3)} = 0.25$. The service rates are $\mu^{(0)} = \mu^{(2)} = 1$ and $\mu^{(1)} = 0.5$. Queue capacities are limited to $N^{(0)} = N^{(1)} = N^{(3)} = 20$. This means that the system as a whole is unstable.

The highest-priority class (class 0) has a small queue because the station is four times faster than the rate of its arrivals. For class 1, the utilization equals one; i.e., the medium priority queue will slowly increase up to the buffer limit. The lowest-priority class must wait for the first and second class to free the space; at the beginning, its mean queue increases, but then this class is gradually eliminated from service as the medium class fills the buffer. The process is presented in Figure 13.

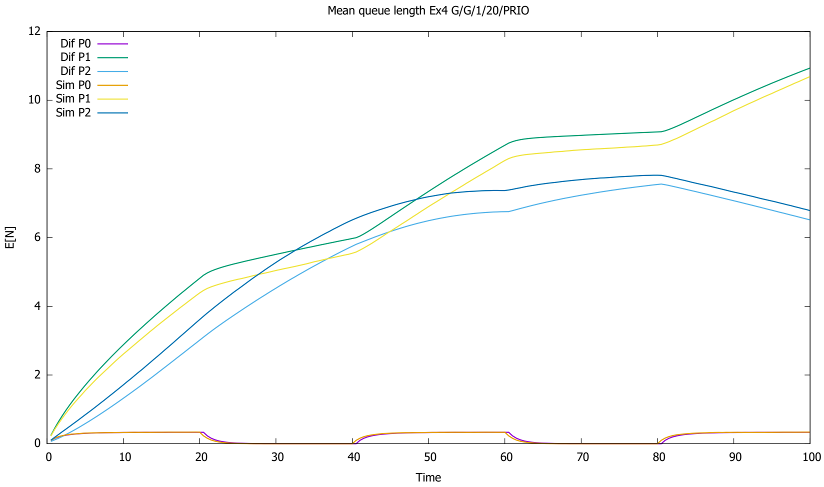

Figure 13. Section 3.4, three priorities: mean queue lengths as a function of time for priority (P0), medium-priority (P1), and low-priority classes.

Figure 14 compares the total number of customers of classes in two groups: (1) high and medium priority and (2) all three priorities together. Both groups contain medium class, which constantly increases and it results in the increase in both total queues.

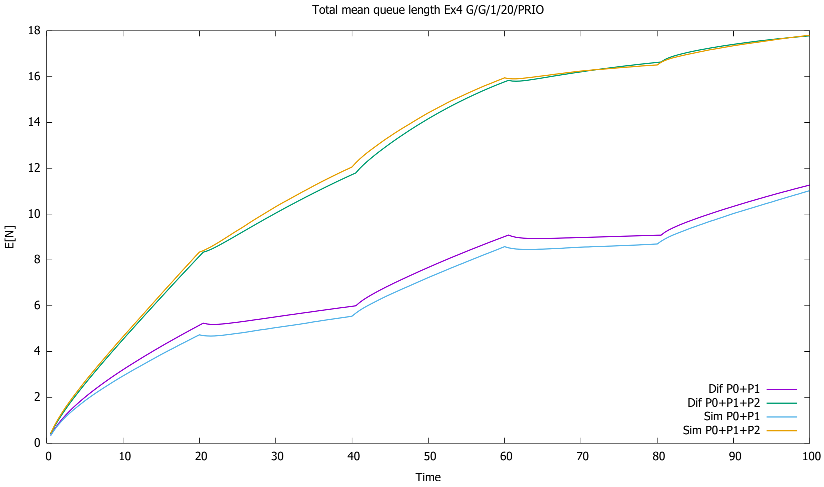

Figure 14. Section 3.4, three priorities: total mean queue length as a function of time of three priority classes.

The probability of the empty queue is close to one for the highest-priority class and changes periodically with the active and non-active traffic intervals of this class. For the other classes, this probability is constantly decreasing, as seen in Figure 15.

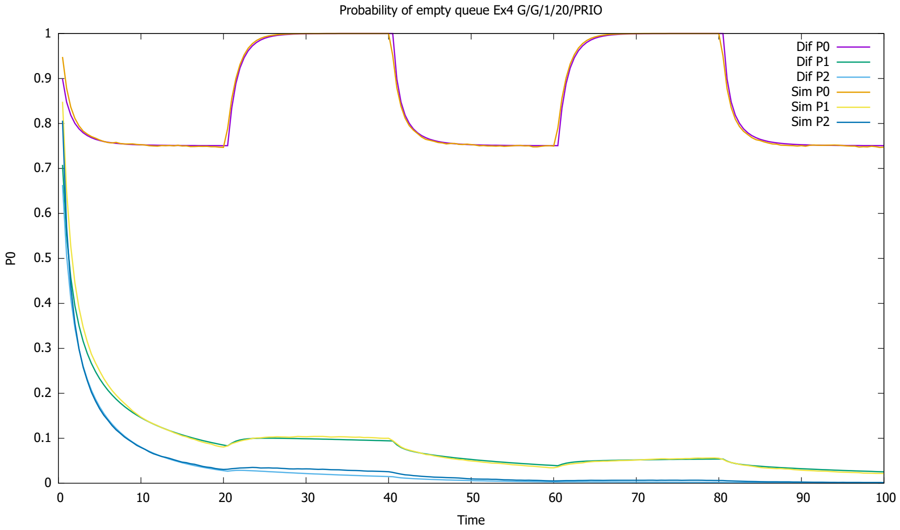

Figure 15. Section 3.4, three priorities; probabilities of empty queues as a function of time for three priority classes (P0), (P1), (P2).

The probability of a full queue is so small for the highest class (P0) that we did not receive it in simulations; see Figure 16. For medium and low priorities, this probability increases with time and is much faster in the case of P(2).

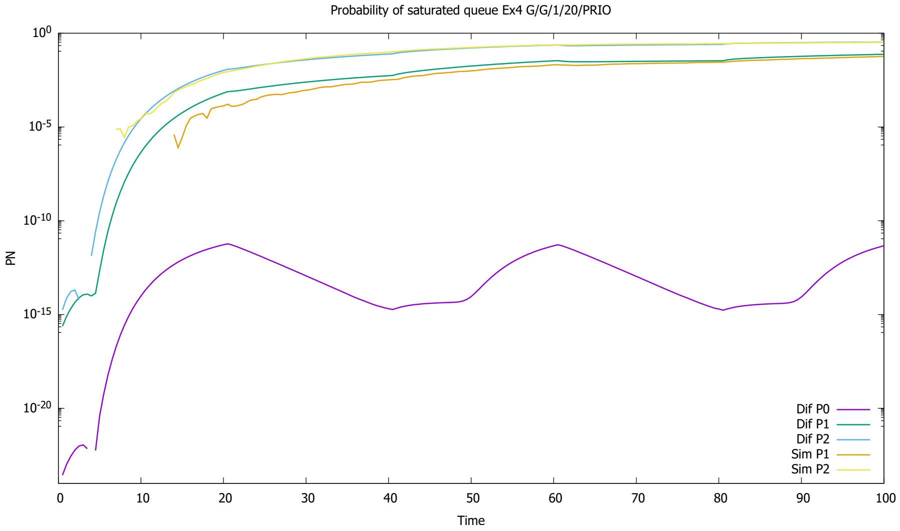

Figure 16. Section 3.4, three priorities: probabilities of saturated queues as a function of time; in case of P(0) only diffusion results are available, the simulations were too short to give such small values.

3.5. Two Priorities; General Interarrival and Service Time Distributions

This example considers a server with two priority levels; the input traffic is non-Poisson for both classes. The priority customers come with intensity $\lambda^{(0)} = 0.75$, and the squared coefficient of variation of their interarrival time distribution is $C_A^{(0)^2} = 8$ during intervals $t \in [0, 20], [40, 60], [80, 100]$. Otherwise, $\lambda^{(0)} = 0$. The intensity of non-priority traffic is constant, $\lambda^{(1)} = 0.75$, with squared coefficient of variation $C_A^{(1)^2} = 5$. The queue capacities are limited to $N^{(0)} = N^{(1)} = 20$.

The service rates are $\mu^{(0)} = \mu^{(1)} = 1$, and the distributions of service time have $C_B^{(0)^2} = 8$, $C_B^{(1)^2} = 5$. Such high values of C_A^2, C_B^2 are not to be observed in real traffic: they are usually below 2. It is known that the errors of the approximation increase with the value of C_A^2, C_B^2 [46,56]; therefore we wanted to check the accuracy of the model for an extreme set of parameters. In diffusion approximation, the type of distributions is not important, only the value of its first two moments. In simulation we used Cox distributions with the same moments. For another distribution, the simulation results would be slightly different. As in previous examples, we present mean queue lengths of priority and non-priority customers (Figure 17), mean queue length of both priority classes together (Figure 18), probabilities of empty queues for priority and non-priority classes (Figure 19), and probabilities of saturated queues for priority and non-priority classes (Figure 20).

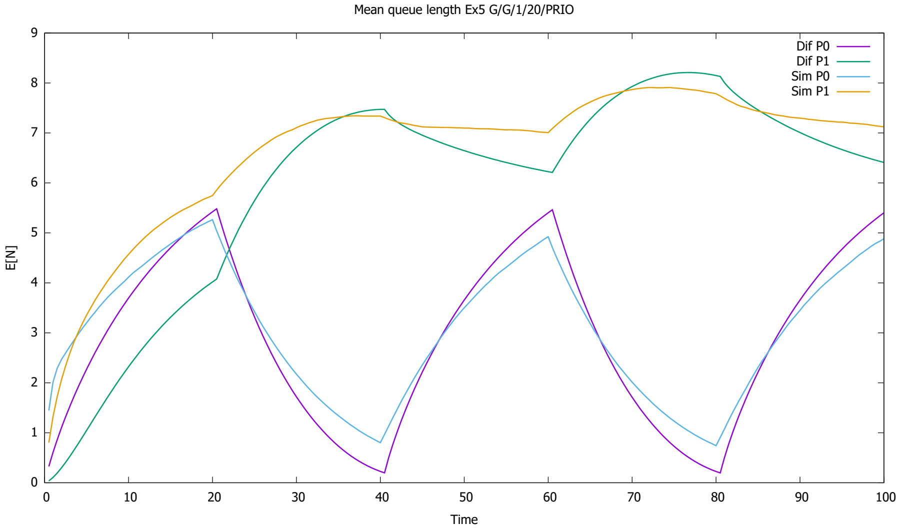

Figure 17. Section 3.5: mean queue lengths of priority (P0) and non-priority (P1) classes for very high values of C_A^2, C_B^2.

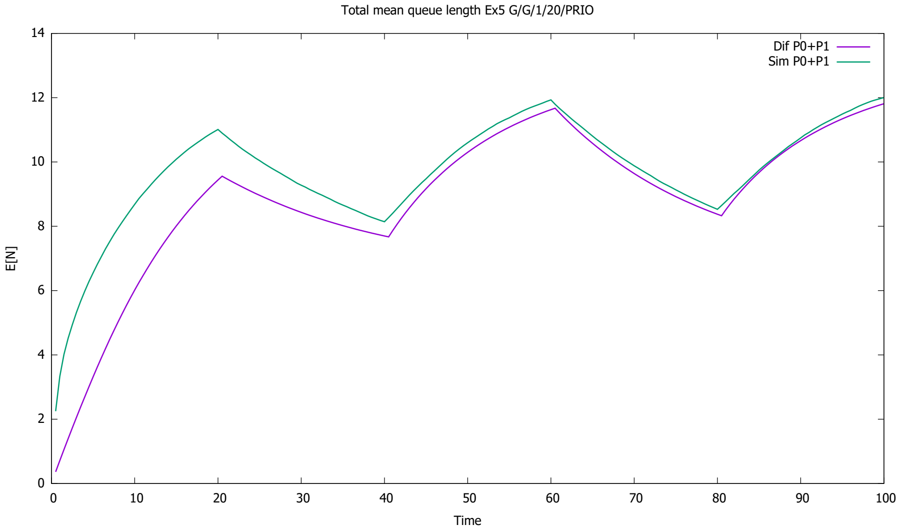

Figure 18. Section 3.5: total mean queue length of both priority classes for very high values of C_A^2, C_B^2.

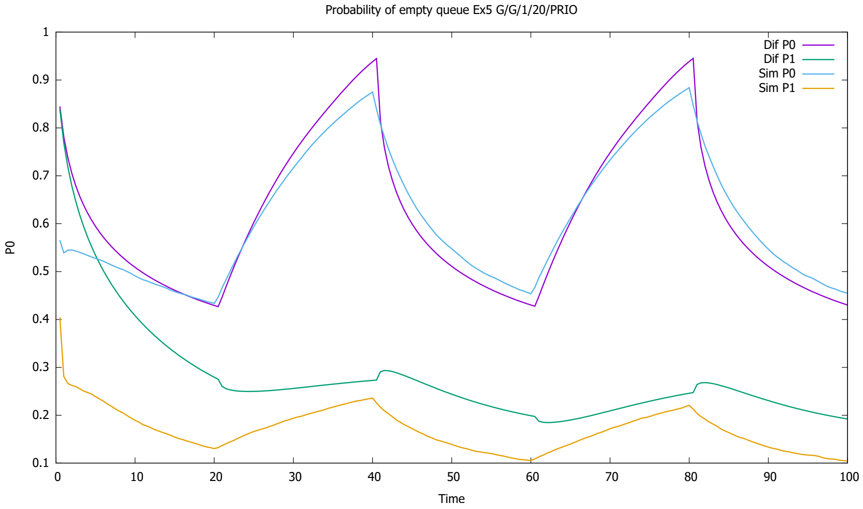

Figure 19. Section 3.5: probabilities of empty queues for priority (P0) and non-priority (P1) classes for very high values of C_A^2, C_B^2.

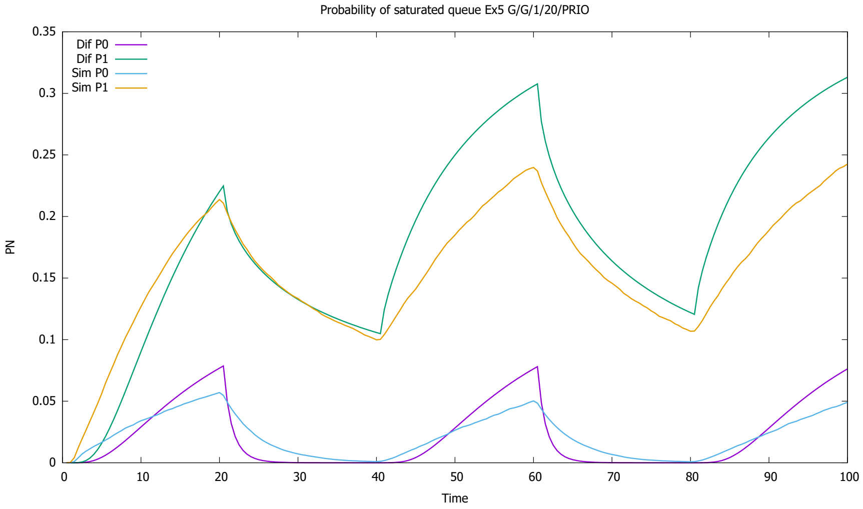

Figure 20. Section 3.5: probabilities of saturated queues for priority (P0) and non-priority (P1) classes for very high values of C_A^2, C_B^2.

The results confirm the deterioration of the approximation: diffusion results are not as close to simulations as was the case previously, when we assumed that $C_A^2 = C_B^2 = 1$; the impact of the squared coefficients of variation is visible. Note also that all model computations are performed inside small time intervals of one unit length, and the approximate distribution of the queue length at the end of one interval gives approximated initial conditions for the next one, increasing the deficiencies of the model. However, the results are still useful in the evaluation of a time-dependent behavior of the system and follow the general pattern given by simulation. A better match is observed for the priority class. It is also natural, as the evolution of the non-priority queue is based on the previous estimation of priority queue, and the errors add up.

The system is slightly unstable; therefore, mean queues slowly increase from one cycle to another (Figure 18). In addition, probabilities of queue saturation increase, as shown in Figure 19, and probabilities of the empty queue decrease with time, as shown in Figure 20.

4. Network of Priority and Non-Priority Queues

The steady-state diffusion model of an open network of G/G/1 or G/G/1/N queues was presented in [53], and it was adapted to transient states in, e.g., [57] and time-dependent routing in [48]. It is here extended to a time-dependent model of a network including both FIFO and priority stations. The approach is based on the decomposition of the network: we need to determine the input flow parameters at each station; then, we may use models of separate stations as discussed in Section 2.

Let M be the number of stations and $L+1$ the number of classes, $k = 0, 1, \dots L$. The traffic intensity $\lambda_i^{(k)}$ of class k at station i is determined by the system of $M(L+1)$ equations representing the balance of flows:

$$\lambda_i^{(k)} = \lambda_{0i}^{(k)} + \sum_{j=1}^{M} \sum_{l=1}^{l=L} \lambda_j^{(l)} r_{ji}^{lk}, \qquad i = 1, \dots, M, \quad l = 0, \dots, L. \tag{17}$$

where r_{ji}^{lk} is the probability that a customer who belongs to station j and to class l goes next to the station i as a class k customer and $\lambda_{0i}^{(k)}$ is an external flow coming to station i.

To obtain the variance of interarrival times at any station i, we have to express it by the variances of interdeparture times, i.e., C_{Dj}^2 or $C_{Dj}^{(l)^2}$ at all stations sending customers to station i. In addition, we need to express the variance of interdeparture times at any station by the variance of interarrival times at the same station.

Both dependencies mutually relating the input and output of the stations turn out to be linear with respect to C_{Dj}^2 and C_{Ai}^2, and the simultaneous solution of the resulting system of equations brings us C_{Ai}^2 or $C_{Ai}^{(k)^2}$.

In transient analysis of the whole network, these equations are to be solved in time intervals that are sufficiently short to consider the flows, routing probabilities, and station utilization as constant parameters. We also distinguish input and output flows of a station; the output is changing continuously with changes in utilization $\varrho_i(t)$.

Assuming that the arrivals to a station i from other stations and from outside the network are independent, and assuming the variances of the arrivals from all directions, we come to the expression (18); see, e.g., Reference [48] for details. The variance of interarrival times at each station is obtained due the equations defining the variance of interdeparture times as a function of the parameters of the interarrival times at each station:

$$C_{Aj}^{(l)^2} = \frac{1}{\lambda_j^{(l)}} \sum_{i=1}^{M} \sum_{k=1}^{L} r_{ij}^{kl} \lambda_i^{(k)} [(C_{Di}^{(k)^2} - 1) r_{ij}^{(kl)} + 1] + \frac{C_{0j}^{(l)^2} \lambda_{0j}^{(l)}}{\lambda_j^{(l)}}, \tag{18}$$

where $C_{0j}^{(l)^2}$ and $\lambda_{0j}^{(l)}$ refer to the flows coming from outside the network to station j as the first station, or for all classes together:

$$C_{Aj}^2 = \frac{1}{\lambda_j} \sum_{l=1}^{L} \lambda_j^{(l)} C_{Aj}^{(l)^2}. \tag{19}$$

The second type of equations linking the variances of $f_{Aj}(x)$ and $f_{Dj}(x)$, where $f_{Dj}(x)$ is the pdf of interdepature times at station j, will be discussed separately for FIFO and priority stations.

4.1. The Output Stream at the FIFO Station

The equations are based on Burke theorem [58]: if a station is active (i.e., it occurs with probability ϱ), the customers leave it in intervals equal to service times; otherwise we should wait for somebody to come and then serve them:

$$f_{Dj}(x) = \varrho_j f_{Bj}(x) + [1 - \varrho_j] f_{Aj}(x) * f_{Bj}(x), \qquad j = 1, \dots, M, \tag{20}$$

where $f_{Aj}(x)$ and $f_{Bj}(x)$ are density functions of interarrival and service times distributions at station j and * is the convolution. If the input flow is not Poisson, the use of interarrival time density $f_{Aj}(x)$ is an approximation; in fact it should be the pdf of idle time distribution. From Equation (20), we obtain

$$C_{Dj}^2 = \varrho_j^2(t) C_{Bj}^2 + C_{Aj}^2(1 - \varrho_j) + \varrho_j[1 - \varrho_j]\,. \tag{21}$$

and

$$C_{Dj}^{(k)^2} = \frac{\lambda_j^{(k)}}{\lambda_j}(C_{Dj}^2 - 1) + 1\,; \tag{22}$$

4.2. The Output Stream at the Priority Station

To use the same as the above schema in the case of priority stations, we need to develop an expression corresponding to Equation (21)—the distribution of interarrival times at each priority level. To simplify the notation, we omit here the index i denoting the station. Let us denote $f_D^{(k)}(x)$ as the pdf of interdeparture times in the stream of class k customers. It can be expressed as

$$\begin{aligned} f_D^{(k)}(x) \quad = \quad & \frac{\varrho^{(k)}}{1 - R^{(k-1)}} c^{(k)}(x) + \left(1 - \frac{\varrho^{(k)}}{1 - R^{(k-1)}}\right) \\ & \times [(1 - R^{(k-)1}) f_A^{(k)}(x) * c^{(k)}(x) \\ & + R^{(k-1)} f_A(k)(x) * \gamma^{(k-1)}(t) * c^{(k)}(x)], \end{aligned} \tag{23}$$

where $R^{(k)} = \sum_{l=0}^{l=k} \varrho^{(l)}$, $\varrho^{(l)} = \lambda^{(l)}/\mu^{(l)}$.

The components of this expression correspond to three situations that are possible after the departure of any customer of class k:

- The next customer in the class k is in the system (this occurs with probability $\frac{\varrho^{(k)}}{1-R^{(k-1)}}$) and will leave it after its completion time;
- There are no customers of this class in the system, and we shall wait for the time described by $F_A^{(k)}(x)$ until it appears and enters the server;
- No customer of class k is present in the system, and a customer of higher class comes before him, so the busy period $\gamma^{(k-1)}$ must first be terminated.

From the above (23), we calculate the squared coefficient of the variation of interdeparture times for each priority customer, which is needed to integrate a single priority station into a network of such stations. The easiest way to obtain the moments of $f_D^{(k)}(x)$ given by Equation (23) is to use its Laplace transform $\bar{f}_D^{(k)}(s)$ and a formula that is valid for any density function $f_X(t)$ and its Laplace transform $\bar{f}_X(s)$

$$\frac{d^n \bar{f}_X^{(s)}}{ds^n}|_{s=0} = -\frac{d^n}{ds^n}\int_0^\infty f_X(x) e^{-sx} dx = \int_0^\infty f_X(x)(-1)^n x^n e^{-sx} dx = (-1)^n E[X^n]\,.$$

The final formula is as follows:

$$C_D^{(k)^2} = \sum_{l=1}^{k} h^{(k,l)} C_A^{(l)^2} + \psi^{(k)} \tag{24}$$

where

$$h^{(k,l)} \quad = \quad \begin{cases} \left(\zeta^{(k,l)} + \dfrac{1 - R^{(k)}}{1 - R^{(k-1)}} R^{(k-1)} g^{(k-1,l)}\right)(\lambda^{(k)})^2, & l < k, \\ \dfrac{1 - R^{(k)}}{1 - R^{(k-1)}}, & l = k, \end{cases}$$

and

$$
\begin{aligned}
\zeta^{(k,l)} &= \frac{\lambda^{(l)}}{\mu^{(k)}(\beta^{(k-1)})^2} + g^{(k-1,l)}\frac{\Lambda^{(k-1)}}{\mu^{(k)}},\\
g^{(k,l)} &= \frac{1}{(\beta^{(k)})^3},
\end{aligned}
$$

$$
\begin{aligned}
\psi^{(k)} &= \chi^{(k)}(\lambda^{(k)})^2 + \frac{1-R^{(k)}}{1-R^{(k-1)}}\Big\{1 + R^{(k-1)}e^{(k-1)}(\lambda^{(k)})^2\\
&\quad +2\varrho^{(k)}\left(1-\frac{\Lambda^{(k-1)}}{\beta^{(k-1)}}\right)+\\
&\quad -\frac{\lambda^{(k)}R^{(k-1)}}{\beta^{(k-1)}}\left[1+2\varrho^{(k)}\left(1-\frac{\Lambda^{(k-1)}}{\beta^{(k-1)}}\right)\right]\Big\}-1,\\
\chi^{(k)} &= \frac{{C_B^{(k)}}^2+1}{(\mu^{(k)})^2}\frac{\Lambda^{(k-1)}}{\beta^{(k-1)}}\left(\frac{\Lambda^{(k-1)}}{\beta^{(k-1)}}-2\right)-\\
&\quad \frac{\Lambda^{(k-1)}}{(\beta^{(k-1)})^2\mu^{(k)}}+e^{(k-1)}\frac{\Lambda^{(k)}}{\mu^{(k)}}\frac{{C_B^{(k)}}^2+1}{(\mu^{(k)})^2},\\
e^{(k)} &= \frac{1}{(\beta^{(k)})^2}-\frac{1}{(\beta^{(k)})^3}\sum_{l=1}^{k}\frac{\varrho^{(l)}}{R^{(k)}}\mu^{(l)}{C_B^{(l)}}^2.
\end{aligned}
$$

The Equation (24) corresponds to (21) in the case of G/G/1/N station: it defines how the variation in the interdeparture times of the class-k customers depends on the variations of the interarrival times of all classes that may influence the output of this class. The parameters of service time distributions are hidden in the coefficients of the equation.

Equations (18) and (24) taken together with (21) or (24) determine the input flow parameters for each class and each station, allowing us to analyze each station separately.

5. The SDN Switch

The SDN switches were modeled recently with the use of diffusion approximation in [47,48,57]. They considered the switch architecture discussed in [59] and simplified it to a single G/G/1/N station. They argued that since the input and output hardware of an SDN switch is fast, the main component of the switch introducing delay and therefore to be modeled, is the queue of packets waiting until the node identifies to which flow they belong and what output port they are to be sent. Suppose that the identification requires a linear search in a flow table with K entries, and T is the constant time to check one entry. Let ϵ be the probability that the router's flow table does not contain the flow rule for a given packet; this will be discovered after going through all K positions, i.e., after time KT. In this case, the service time is constant, with zero variance. Otherwise, with probability $(1-\epsilon)$, the time to find the existing entry is uniformly distributed in $[T, KT]$ and has a mean $(K+1)T/2$ and variance $(K^2-1)T^2/12$.

In the cited models, if a packet is not identified, it disappears. Here, we follow its way to the controller and its return to the switch via uplink and downlink channels as well as its second service in the switch as a priority customer, similarly as is done in [28], but considering transient behavior of the system and general interarrival and service time distributions. The model gives us a chance to see the delays introduced by the communication with the controller and priority service of returning packets. We may study the behavior of the system as a function of its parameters, such as speed of the switch, the controller and communication channels, and hit ratio for the identification of incoming packets. This system is presented in Figure 21. The model is composed of four service stations: the switch is a G/G/1/N/Priority station, and other stations are modeled as

G/G/1/N. We use Equations (17), (18), (21), and (24) to separate the stations; the structure of the network is simple and these general formulas are thereby greatly simplified.

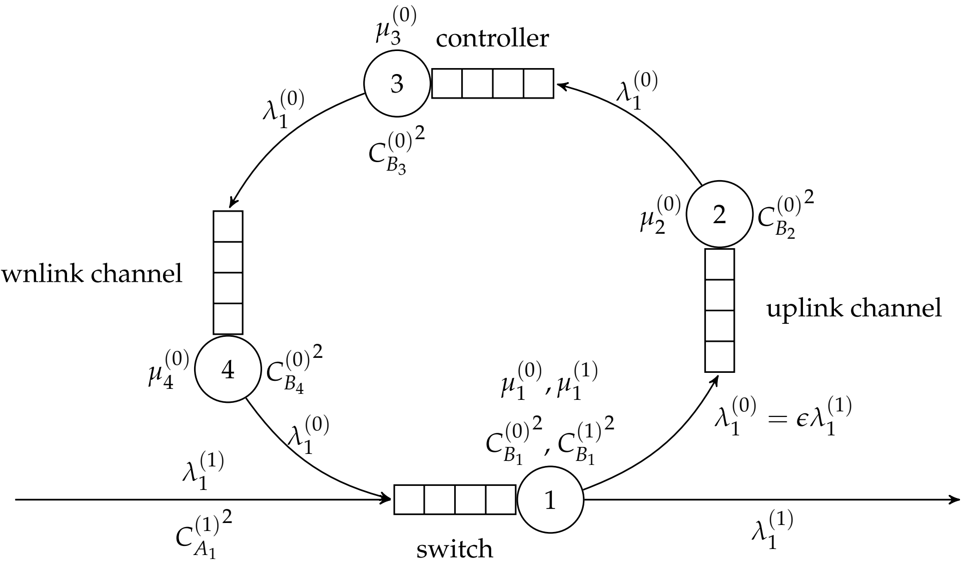

Figure 21. Model of SDN switch and its interactions with the SDN controller.

The length of the diffusion interval, i.e., the maximum size of the queue in the model, significantly affects the numerical solution time of the diffusion equation; the longer the interval, the greater the calculation time. To ease the calculations, we assume that the maximum volume of the switch buffer is $N = 50$ packets when $\epsilon = 0.2$, but when $\epsilon = 0.5$, this means the congestion is higher and queues are longer, and we assume that $N = 100$. The maximum size of other queues is $N = 20$ packets.

In the numerical example below, the changes in the input flow are displayed in Figure 22. They cover an interval of 1 s. We used parameters $K = 950$, $T = 8 \times 10^{-7}$ s (giving $\mu_1 \approx 2630$ packets/s) to determine the distribution of service time at the switch and two values of the probability ϵ of missing a flow description. With this probability, a packet goes (only once) along the loop S2-S3-S4 and comes back to the switch S1 as a priority packet. We assumed for channels; i.e., stations S2, S4, $\mu_2^{(0)} = \mu_4^{(0)} = 1000$ packets/s, and $\mu_3^{(0)} = 1500$ packets/s for the controller. $\epsilon = 0.2$, 0.5. The service time is either constant or uniformly distributed only for nonpriority packets; for priority packets, the distribution is uniform ($\epsilon = 0$ in this case).

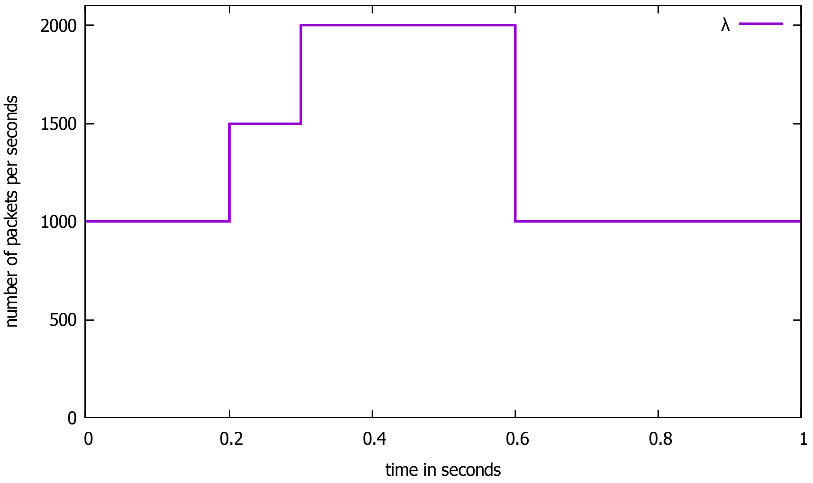

Figure 22. The input flow λ (packets per second) to the SDN switch, considered interval of 1 s.

The transient solution of diffusion equations is computed in time-intervals of the length 5 ms; i.e., we have 200 intervals with constant but different diffusion parameters following the state of the system. Inside an interval, diffusion parameters in single station models are constant; at the end of each interval, the Equations (17), (18), (21), and (24) furnish new traffic parameters for the the diffusion models at the next interval. The queue distributions at the end of an interval are used as initial conditions for the next one.

Below, a few figures illustrate the numerical results. Figure 23 displays the mean queue lengths at the switch for priority (P0) and non-priority (P1) packets as a function of time, reacting to the changes in the input traffic, for $\epsilon = 0.2$. We used a logarithmic scale to show together the results for both priority and non-priority classes, which have significantly different values. The simulation and diffusion approximation results are displayed together. Figure 24 presents similar results for $\epsilon = 0.5$. Comparing both figures, we see the impact of ϵ on the queues; its increase from 0.2 to 0.5 makes the switch maximum mean queue almost ten times longer.

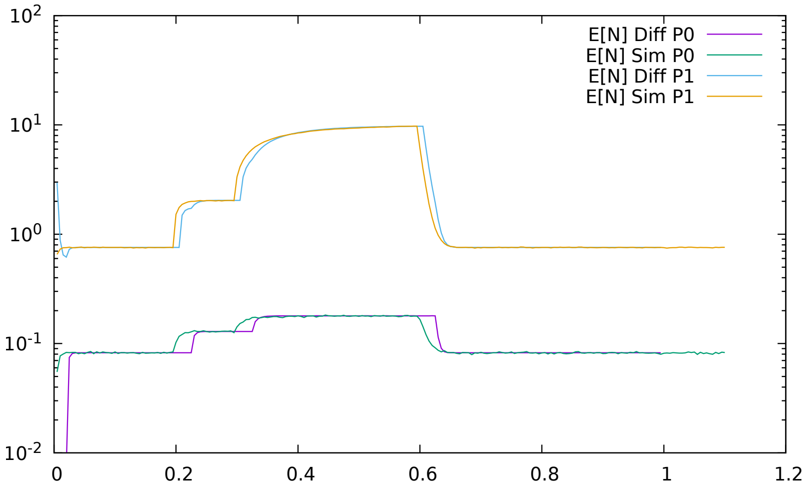

Figure 23. Mean queue length at the switch for priority (P0) and non−priority (P1) packets as a function of time, $\epsilon = 0.2$, diffusion, and simulation results.

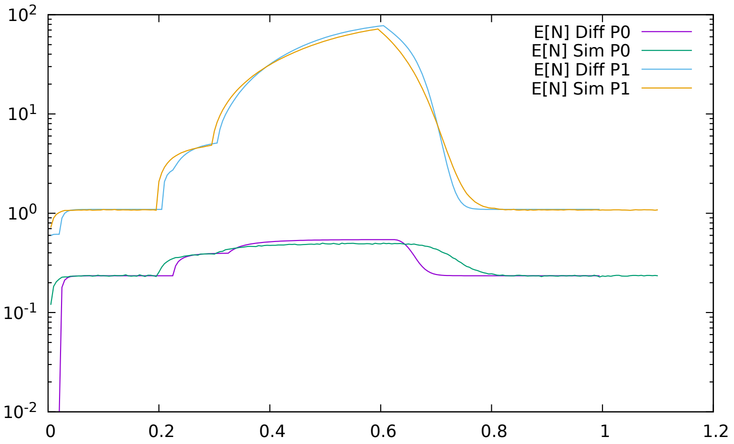

Figure 24. Mean queue length at the switch for priority (P0) and non−priority (P1) packets as a function of time, $\epsilon = 0.5$, diffusion and simulation results.

The same may be observed in Figures 25 and 26, presenting mean response time of the switch as a function of time, for $\epsilon = 0.2$ and $\epsilon = 0.5$, diffusion, and simulation results. The change in ϵ greatly influences the delays. The spikes at the moments of traffic changes come from the fact that we used the Little's formula $E[R] = E[N]/\lambda$, which is correct at steady-state analysis but approximate in the transient one, to obtain the mean response time.

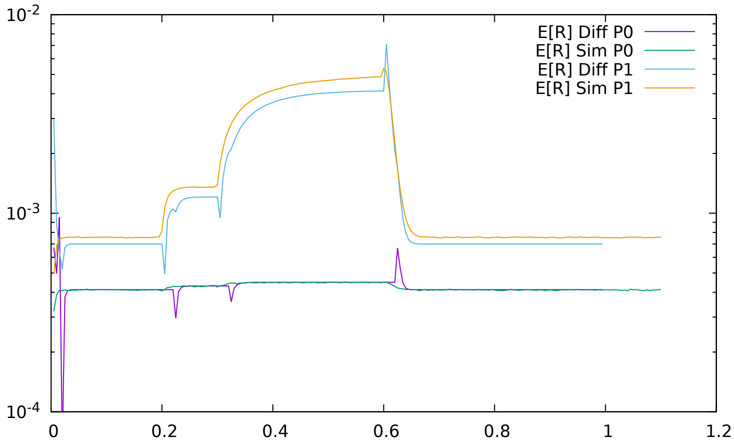

Figure 25. Mean response time at the switch as a function of time, $\epsilon = 0.2$, diffusion, and simulation results.

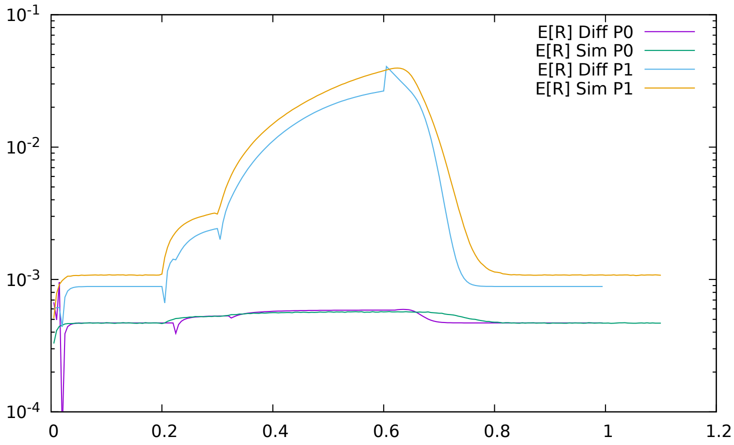

Figure 26. Mean response time at the switch as a function of time, $\epsilon = 0.5$, diffusion, and simulation results.

Figures 27 and 28 give the time-dependent mean delay introduced by the communication with the controller, i.e., the summary mean response time of uplink and downlink channels and the controller, $E[R_2] + E[R_3] + E[R_4]$, after which the packets of previously unrecognized destination come back to the switch together with their flow details. The entire mean response time of the system is

$$E[R] = E[R_1^{(1)}] + (1-\epsilon)(E[R_2] + E[R_3] + E[R_4] + E[R_1^{(0)}]).$$

and the pdf of the R is

$$f_R(x) = f_{R_1^{(1)}}(x) + (1-\epsilon)\Big(f_{R_2}(x) * f_{R_3}(x) * f_{R_4}(x) * f_{R_1^{(0)}}(x)\Big).$$

Figures 29 and 30 present loss probability due to the queue saturation as a function of time, respectively, for $\epsilon = 0.2$ and $\epsilon = 0.5$, obtained by diffusion approximation and simulation. It is visible that 100,000 simulation runs are not enough to obtain reliable results; they are incomplete and mostly nonexistent, while diffusion approximation has no difficulties in modeling very small probability values.

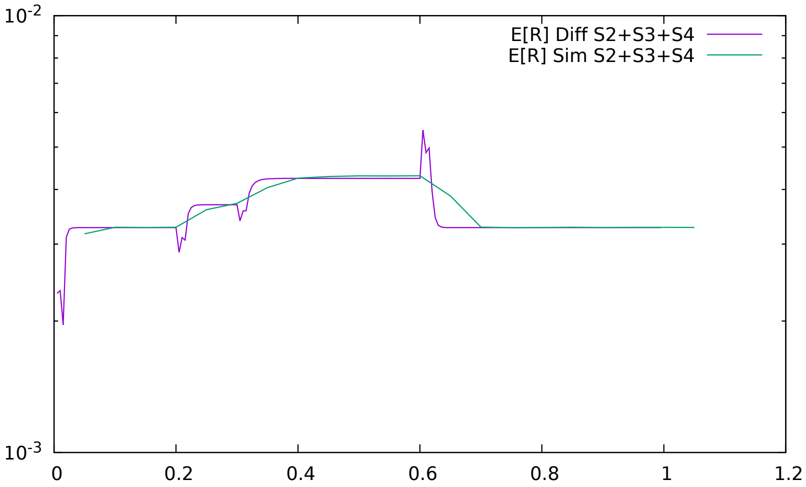

Figure 27. Mean delay introduced by the communication with the controller as a function of time, $\epsilon = 0.2$, diffusion, and simulation results.

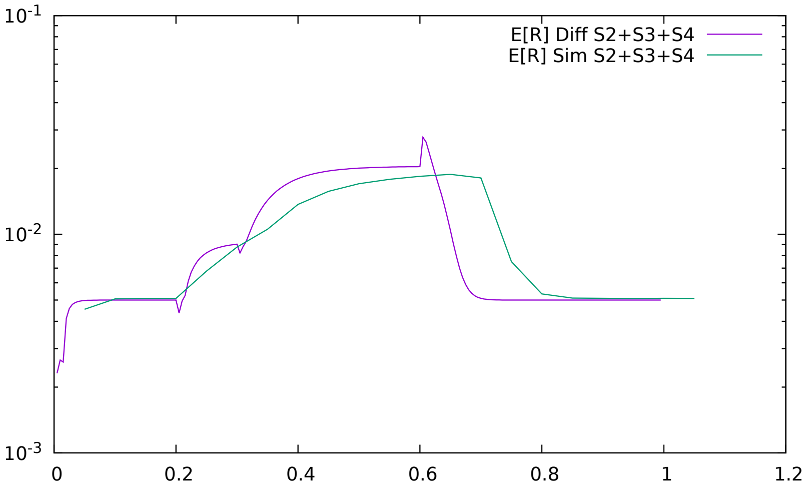

Figure 28. Mean delay introduced by the communication with the controller as a function of time, $\epsilon = 0.5$, diffusion, and simulation results.

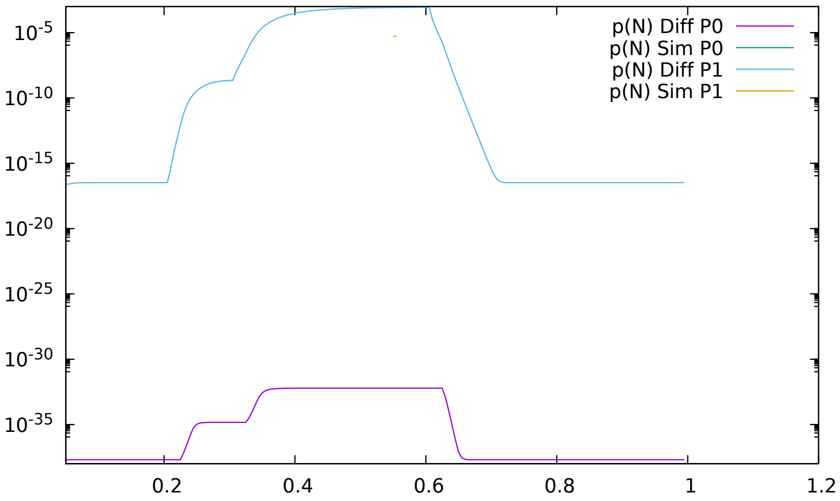

Figure 29. Loss probability due to the queue saturation as a function of time, $\epsilon = 0.2$, diffusion, and simulation results.

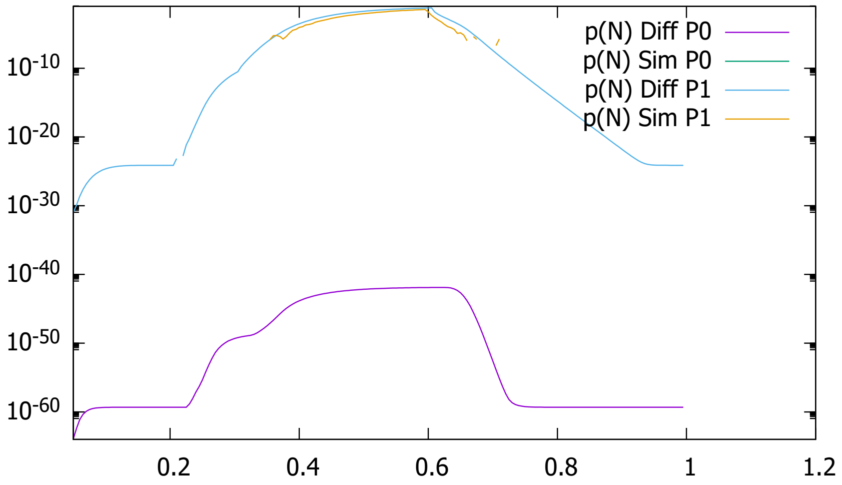

Figure 30. Loss probability due to the queue saturation as a function of time, $\epsilon = 0.5$, diffusion, and simulation results.

The comparison of diffusion and simulation results gives us an estimation of errors introduced by the method, and we conclude that their size is acceptable in general. In addition, the dynamics of changes follows well the one observed in the simulation model. If the controller can change routing each 100 ms, the switch and a network of switches will operate in a transient regime for most of the time. Therefore, every performance evaluation or an optimization study should take a transient analysis into account. Diffusion approximation proves to be a convenient tool for this purpose.

Our original contributions are the following:

- Proposing the diffusion model of a multiclass G/G/1/N/Priority station, i.e., a station with general interarrival and service time distributions, limited buffer, and with preemptive-resume priority queues. Each class of customers has its specified priority level and its own parameters of the interarrival and service time distribution. Within one priority class, the scheduling is based on the FIFO algorithm. The model covers transient and steady-state analysis.
- Validation of this model by comparison with discrete-event simulation for various loads and interarrival and service time distributions, and discussion of errors;
- The model of an open network with any topology integrationg priority and FIFO service stations;
- A model of SDN switch exchanging packets with undetermined routing with SDN controller and validation of this model.

General distributions, priority classes, transient analysis, flexible topology, the form of results which is not restricted to mean values but giving the distributions, make the proposed model broader than the existing ones.

6. Conclusions

The article proposes a queueing model of G/G/1/N/preemptive-resume priority station serving customers with any number of priority classes. The main features of this model are general distributions of interarrival and service times and transient analysis of the queues. The results of a single station model were verified and validated in detail by comparison with simulations for various patterns of time-dependent traffic. In addition, the results of the ring switch–uplink–controller–downlink were verified with the simulations. In most cases, the approximation quality is very good, especially when the squared coefficient of variation of interarrival and service time distributions is close to one. The factors negatively affecting the approximation are: the increasing number of priority levels, because the results for a certain class depend on results (and errors) for all higher classes, and very large variances of interarrival and service time distributions. Furthermore, the results for the network are worse than for a single station: the errors of determination of variation of flows and the errors of dynamics prediction in intermediate stations (uplink, controller, downlink) bring additional errors into the switch model.

The model gives an insight into the performance of a priority service station. The impact of the utilization of the system on queue lengths and response times at various priority levels is visible. A network model integrating any number of G/G/1/N/preemptive-resume and G/G/1/N stations, both for the steady-state and transient regime, is presented and used to study the performances of SDN switch receiving a flow of packets with the intensity, which is frequently changing due to the decision SDN controller. The model includes the communication between switch and controller for packets belonging to flows unrecognized by the switch. It may be used to study the impact of the speed of switch, controller, and communication between them on performances of the SDN network, including such quality of service factors as delay, jitter, and losses. It also allows us to evaluate the effect of the hit ratio (probability that a packet belongs to a flow that is known to the switch) on the switch response time, the possible starvation problem on the lower priority level, and loss probabilities of packets. Another advantage of the diffusion approach is that it gives us the distributions of queues and delays, not only their mean values. It also means that we determine the probability that a packet is lost because of the saturation of the buffer. The obtained numerical results indicate that the transient regime may take a significant part of the total switch operation time; therefore, the diffusion approach to study transient periods is fully justified. In future work, we will focus on validating the model of the entire SDN network with any number of switches.

Author Contributions: T.C. proposed the algorithm to sole transient state diffusion models and the initial version of the priority model, T.N. corrected the priority model and performed numerical computations and simulations, M.N. wrote a part of the text, supervised the whole process, and edited the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Anerousis, N.; Chemouil, P.; Lazar, A.A.; Mihai, N.; Weinstein, S.B. The Origin and Evolution of Open Programmable Networks and SDN. *IEEE Commun. Surv. Tutor.* **2021**. [CrossRef]
2. Benzekki, K.; Fergougui, A.E.; Elalaoui, A.E. Software-defined networking (SDN): A survey. *Secur. Commun. Netw.* **2016**, *9*, 5803–5833. [CrossRef]
3. Shirmarz, A.; Ghaffari, A. Performance issues and solutions in SDN-based data center: A survey. *J. Supercomput.* **2020**, *76*, 7545–7593. [CrossRef]
4. Thirupathi, V.; Sandeep, C.; Kumar, S.N.; Kumar, P.P. A Comprehensive Review on SDN Architecture, Applications And Major Benifits of SDN. *Int. J. Adv. Sci. Technol.* **2019**, *28*, 607–614.
5. Agg, P.; Johanyák, Z.C.; Szilveszter, K. Survey on SDN Programming Languages. In Proceedings of the 8th International Scientific and Expert Conference TEAM 2016, Rome, Italy, 24–26 February 2016; pp. 64–70.
6. Mahmood, W.; Nasir, S.D.; Waqas, I. A Research Survey on Software Defined Networking (SDN). In Proceedings of the Ninth International Conference on Advances in Computing, Control and Networking (ACCN 2019), London, UK, 21 July 2019; pp. 127–132. [CrossRef]
7. Yu, T.; Hong, Y.; Cui, H.; Jiang, H. A survey of Multi-controllers Consistency on SDN. In Proceedings of the 2018 4th International Conference on Universal Village (UV), Boston, MA, USA, 21–24 October 2018; pp. 1–6. [CrossRef]
8. Stevens, M.; Ng, B.; Streader, D.; Welch, I. Global and local knowledge in SDN. In Proceedings of the 2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 18–20 November 2015; pp. 237–243. [CrossRef]
9. Paliwal, M.; Shrimankar, D.; Tembhurne, O. Controllers in SDN: A Review Report. *IEEE Access* **2018**, *6*, 36256–36270. [CrossRef]

10. Michel, O.; Keller, E. SDN in wide-area networks: A survey. In Proceedings of the 2017 Fourth International Conference on Software Defined Systems (SDS), Valencia, Spain, 8–11 May 2017; pp. 37–42. [CrossRef]
11. Das, T.; Sridharan, V.; Gurusamy, M. A Survey on Controller Placement in SDN. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 472–503. [CrossRef]
12. Isong, B.; Molose, R.R.S.; Abu-Mahfouz, A.M.; Dladlu, N. Comprehensive Review of SDN Controller Placement Strategies. *IEEE Access* **2020**, *8*, 170070–170092. [CrossRef]
13. Mbodila, M.; Isong, B.; Gasela, N. A Review of SDN-Based Controller Placement Problem. In Proceedings of the 2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Kimberley, South Africa, 25–27 November 2020; pp. 1–7. [CrossRef]
14. Yan, B.; Liu, Q.; Shen, J.; Liang, D.; Zhao, B.; Ouyang, L. A survey of low-latency transmission strategies in software defined networking. *Comput. Sci. Rev.* **2021**, *40*, 100386. [CrossRef]
15. Yang, L.; Ng, B.; Seah, W.K.; Groves, L.; Singh, D. A survey on network forwarding in Software-Defined Networking. *J. Netw. Comput. Appl.* **2021**, *176*, 102947. [CrossRef]
16. Hemanth, D.J. A Survey on Traffic Prediction and Classification in SDN. *Intell. Syst. Comput. Technol.* **2020**, *37*, 367–370. [CrossRef]
17. Priyadarsini, M.; Bera, P. Software defined networking architecture, traffic management, security, and placement: A survey. *Comput. Netw.* **2021**, *192*, 108047. [CrossRef]
18. Moin, S.; Karim, A.; Safdar, K.; Iqbal, I.; Safdar, Z.; Vijayakumar, V.; Ahmed, K.T.; Abid, S.A. GREEN SDN—An enhanced paradigm of SDN: Review, taxonomy, and future directions. *Concurr. Comput. Pract. Exp.* **2018**, *32*, 1–21. [CrossRef]
19. Rasool, Z.I.; Ali, R.S.A.; Abdulzahra, M.M. Network Management in Software-Defined Network: A Survey. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *1094*, 1–6. [CrossRef]
20. Ahmad, S.; Mir, A.H. Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers. *J. Netw. Syst. Manag.* **2021**, *29*. [CrossRef]
21. Sandhya; Sinha, Y.; Haribabu, K. A survey: Hybrid SDN. *J. Netw. Comput. Appl.* **2017**, *100*, 35–55. [CrossRef]
22. Khorsandroo, S.; Sánchez, A.G.; Tosun, A.S.; Arco, J.; Doriguzzi-Corin, R. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. *Comput. Netw.* **2021**, *192*, 107981. [CrossRef]
23. Keshari, S.K.; Kansal, V.; Kumar, S. A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN). *Wirel. Pers. Commun.* **2021**, *116*, 2593–2614. [CrossRef]
24. Mahmood, K.; Chilwan, A.; Østerbø, O.; Jarschel, M. Modelling of OpenFlow-based software-defined networks: The multiple node case. *IET Netw.* **2015**, *4*, 278–284. [CrossRef]
25. Ansell, J.; Seah, W.K.G.; Ng, B.; Marshall, S. Making Queueing Theory More Palatable to SDN/OpenFlow-based Network Practitioners. In Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; IEEE: Istanbul, Turkey, 2016; pp. 1119–1124. [CrossRef]
26. Sood, K.; Yi, S.; Xiang, Y. Performance Analysis of Software-Defined Network router using M/Geo/1. *IEEE Commun. Lett.* **2016**, *20*, 27–51. [CrossRef]
27. Miao, W.; Min, G.; Wu, Y.; Wang, H. Performance Modelling of Preemption-Based Packet Scheduling for Data Plane in Software Defined Networks. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 60–65. [CrossRef]
28. Miao, W.; Min, G.; Wu, Y.; Wang, H.; Hu, J. Performance Modelling and Analysis of Software-Defined Networking under Bursty Multimedia Traffic. *ACM Trans. Multimed. Comput. Commun. Appl.* **2016**, *12*, 24–36. [CrossRef]
29. Singh, D.; Ng, B.; Lai, Y.C.; Lin, Y.D.; Seah, W.K.G. Modelling Software-Defined Networking: Software and Hardware Switches. *J. Comput. Netw. Comput. Appl.* **2018**, *122*, 24–36. [CrossRef]
30. Goto, Y.; Ng, B.; Seah, W.K.G.; Takahashi, Y. Queueing analysis of software defined network with realistic OpenFlow-based switch model. *Comput. Netw.* **2019**, *164*, 106892. [CrossRef]
31. Mochalov, V.P.; Linets, G.I.; Palkanov, I. The Erlang Model for a Fragment of SDN Architecture. In *Advances in Automation II, Proceedings of the RusAutoConf 2020, Sochi, Russia, 6–12 September 2020*; Lecture Notes in Electrical Engineering; Springer International Publishing: Cham, Switzerland, 2021; Volume 729, pp. 424–437. [CrossRef]
32. Azodolmolky, S.; Wieder, P.; Yahyapour, R. Performance evaluation of a scalable software-defined networking deployment. In Proceedings of the 2013 Second European Workshop on Software Defined Networks, Berlin, Germany, 10–11 October 2013; IEEE: Berlin, Germany, 2013; pp. 68–74. [CrossRef]
33. Azodolmolky, S.; Nejabati, R.; Pazouki, M.; Wieder, P.; Yahyapour, R.; Simeonidou, D. An analytical model for software defined networking: A network calculus-based approach. In Proceedings of the IEEE Global Communications Conference, Atlanta, GA, USA, 9–13 December 2013; IEEE: Atlanta, GA, USA, 2013; pp. 1397–1402. [CrossRef]
34. Bozakov, Z.; Rizk, A. Taming SDN controllers in heterogeneous hardware environments. In Proceedings of the 2013 Second European Workshop on Software Defined Networks, Berlin, Germany, 10–11 October 2013; IEEE: Berlin, Germany, 2013; pp. 50–55. doi:10.1109/EWSDN.2013.15. [CrossRef]
35. Champernowne, D.C. An elementary method of solution of the queueing problem with a single server and constant parameters. *J. R. Statist. Soc.* **2000**, *3*, 263–266. [CrossRef]
36. Takâcs, L. *Introduction to the Theory of Queues*; Oxford University Press: Oxford, UK, 1960.
37. Tarabia, A.M.K. Transient Analysis of M/M/1/N Queue—An Alternative Approach. *Tamkang J. Sci. Eng.* **2000**, *3*, 263–266.

38. Parthasarathy, P.; Selvaraju, N. Transient analysis of a queue where potential customers are discouraged by queue length. *Math. Probl. Eng.* **2001**, *7*, 433–454. [CrossRef]
39. Parthasarathy, P.; Sudhesh, R. Exact transient solution of a discrete time queue with state-dependent rates. *Am. J. Math. Manag. Sci.* **2006**, *26*, 253–276. [CrossRef]
40. Sudhesh, R. Transient analysis of a queue with system disasters and customer impatience. *Queueing Syst.* **2010**, *66*, 95–105. [CrossRef]
41. Vuppalapati, N.; Venkatesh, T.G. Modeling & analysis of software defined networks under non-stationary conditions. *Peer Netw. Appl.* **2021**, *14*, 1174–1189. [CrossRef]
42. Tipper, D.; Sundareshan, M. Numerical methods for modeling computer networks under nonstationary conditions. *IEEE J. Sel. Areas Commun.* **1990**, *8*, 1682–1695. [CrossRef]
43. Misra, V.; Gongnad, W.B.; Towsley, D. A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM 2000), Stockholm, Sweden, 28 August–1 September 2000; ACM: New York, NY, USA, 2000; pp. 152–160. [CrossRef]
44. Gelenbe, E. On Approximate Computer Systems Models. *J. ACM* **1975**, *22*, 261–269. [CrossRef]
45. Reinecke, P.; Krauß, T.; Wolter, K. HyperStar: Phase-Type Fitting Made Easy. In Proceedings of the 9th International Conference on the Quantitative Evaluation of Systems (QEST 2012), London, UK, 17–20 September 2012; pp. 201–202.
46. Czachórski, T.; Nycz, M.; Nycz, T. Modelling transient states in queueing models of computer networks: A few practical Issues. In *Distributed Computer and Communication Networks*; Vishnevsky, V., Ed.; Springer: Moscow, Russia, 2013; Volume 279, pp. 58–72. [CrossRef]
47. Czachórski, T.; Gelenbe, E.; Suila, K.; Marek, D. Transient behaviour of a network router. In Proceedings of the 43th International Conference on Telecommunications and Signal Processing (TSP), Milan, Italy, 7–9 July 2020; pp. 246–252. [CrossRef]
48. Czachórski, T.; Gelenbe, E.; Kuaban, G.S.; Marek, D. Time-Dependent Performance of a Multi-Hop Software Defined Network. *Appl. Sci.* **2021**, *11*, 2469. [CrossRef]
49. Czachórski, T.; Nycz, T.; Pekergin, F. Transient States of Priority Queues—A Diffusion Approximation Study. In Proceedings of the Fifth Advanced International Conference on Telecommunications, AICT 2009, Venice/Mestre, Italy, 24–28 May 2009; pp. 44–51. [CrossRef]
50. 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.
51. Czachórski, T. Queuing models for performance evaluation of computer networks: Transient state analysis. In *Analytic Methods in Interdisciplinary Applications*; Mityushev, V., Ruzhansky, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 116, pp. 51–80. [CrossRef]
52. Cox, R.P.; Miller, H.D. *The Theory of Stochastic Processes*; Chapman and Hall: London, UK, 1965.
53. Gelenbe, E.; Pujolle, G. The behaviour of a single queue in a general queueing network. *Acta Inform.* **1976**, *7*, 123–136. [CrossRef]
54. Jaiswal, N.K. *Priority Queues*, 1st ed.; Academic Press: New York, NY, USA, 1968.
55. OMNET++ Community Site. Available online: http://www.omnetpp.org/ (accessed on 27 May 2021).
56. Czachórski, T.; Pekergin, F. Diffusion approximation as a modelling tool. In *Network Performance Engineering*; Kouvatsos, D.D., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 5233, pp. 447–476. [CrossRef]
57. Czachórski, T.; Gelenbe, E.; Kuaban, G.S.; Marek, D. Time Dependent Diffusion Model for Security Driven Software Defined Networks. In Proceedings of the Second International Workshop on Stochastic Modeling and Applied Research of Technology (SMARTY 2020), CEUR-WS, Petrozavodsk, Russia, 16–20 August 2020; Volume 2792, pp. 38–56.
58. Burke, P.J. The Output of a Queuing System. *Oper. Res.* **1956**, *4*, 699–704. [CrossRef]
59. Wijeratne, S.; Ekanayake, A.; Jayaweera, S.; Ravishan, D.; Pasqual, A. Scalable High Performance router Architecture on FPGA for Core Networks. In Proceedings of the 2019 ACM/SIGDA International Symposium, Seaside, CA, USA, 24–26 February 2019; ACM: New York, NY, USA, 2019. [CrossRef]