Software Defined Network Dynamics via Diffusions

Tadeusz Czachórski[1], Erol Gelenbe[1], and
Dariusz Marek[2]

[1] Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Bałtycka 5, 44-100 Gliwice, Poland
[2] The Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland

Abstract. Software-Defined Networks (SDN) dynamically modify the paths of Internet flows in response to the quality of service or security needs, and hence frequently modify traffic levels at network routers. Thus network routers often operate in the transient regime, rather than at steady-state, with significant impact on packet loss probabilities and delay. We, therefore, investigate the time-dependent performance of a small network of routers, modelled as G/G/1/N queueing stations. A diffusion approximation is developed to predict the quality of service of the routers in the transient regime. Numerical examples show that the results in the transient regime can differ very significantly from the steady-state results, and therefore that the transient analysis must be taken into account in evaluating the performance of routers in a SDN network.

1 Introduction

The performance of computer networks since its beginnings, was investigated [1, 2] via networks of node queues that contain packets, representing interconnected routers that forward packets from source to a destination over several hops or routers. These models are used to compute in steady-state, the network delays and loss probabilities in routers, and to predict or optimise the overall transmission quality of service. The evolution of computer networks has resulted in new architectures, methods and models have been adapted and new parameters introduced.

The increased use of the Internet to carry voice traffic [3], as well as the Internet of Things, Cloud, Fog and Edge computing [4] brings new challenges by increasing the variety of network architectures and the complex stochastic nature of the transmitted flows. Also, the increased use of SDN controllers inside networks [5–7] creates frequent changes in traffic patterns and paths, and hence dynamic changes also in the traffic intensity of different paths and of the traffic carried by routers in the network.

These changes raise a new need for transient analysis since the network is seldom operating at "steady-state". In this context, queueing networks [1] diffusion approximations [8], fluid approximations [9], and network calculus [10, 11] are some of the mathematical techniques that may be used.

Earlier analytical studies of SDN networks are based on steady-state models based on discrete Markov chain analysis [12–17] with Poisson flows, or on network calculus [18, 19] which only provides a rough estimate of network delays, especially when frequent controller decisions modify the traffic load. To address this concern, in recent work, we have considered a single SDN forwarder and modelled it with diffusion approximations [20] to determine its transient behaviour. The accuracy of the diffusion approximation has been examined in several studies such as [21], and was found to be acceptable with regard to other forms of analysis such as discrete event simulations which are subject to statistical confidence intervals and long simulation runs. Therefore in this paper, we will not present comparisons of the diffusion results with discrete event simulation.

The large scale deployment of the IoT [22] together with Cloud Services [4], where measured data from sensors is transported to the cloud for decision making and the control of cyber-physical systems, has increased the complexity and challenges of networks that must now ensure better security [23] of the traffic flows, acceptable quality of service, flexible network management, and energy optimisation [24]. The introduction of artificial intelligence into network routing and management [25, 26] also provides greater variability in the flows that traverse the network. Besides, the decoupling of the data plane, the control plane and the application plane through Software Defined Networks (SDN) [5] gives carriers, service providers and enterprises more significant control over the way traffic moves around networks [27, 28] and simplifies network operation, management and administration. However, it imposes frequent updates of network paths and of the traffic levels that are carried by the routers, so that the transient behaviour of network components becomes of great interest.

Recent studies have analyzed SDN networks to optimize steady-state performance using queueing [12–17, 29] and network calculus [18, 19, 30]. However, SDN Controllers modify the state of the network quite frequently, and these issues require the analysis of the transient behaviour of a router. However, the usual tools for network performance are not well adapted to this requirement since the transient analysis of queueing network models is particularly difficult, and the discrete event simulation of the transient behaviour of networks is very time-consuming due to the large number of randomised repetitions that are needed to achieve a reasonable level of statistical accuracy.

Therefore, in this paper, we address the time-dependent behaviour of a router using diffusion approximations which offer two essential advantages: packet interarrival and service times distributions do not depend on the usual "Poisson and exponential" assumptions, and they lead to computationally efficient results concerning the system's transient behaviour. Additionally, the diffusion model only requires the first two moments of the interarrival and service times, so that relatively realistic parameters can be based on measured traffic data, and

it provides numerical results which are difficult to obtain with other techniques [31]. While the approach for steady-state distributions was introduced decades ago [32, 33] and applied to numerous problems, including to admission control in industrial telecommunication systems [34], and active queue management for non-integer PID controllers in IP routers [35], the transient analysis is more challenging and requires carefully crafted analtical techniques which we use in this paper.

The features which are in favour of the method are that the diffusion model of a single server allows general interarrival and service time distributions for realistic network data, going beyond conventional discrete Markov models. Results are obtained as queue length and waiting time distributions, simplifying the analysis of QoS parameters such as delay, jitter and loss probability. Also, networks may be hierarchical with any topology and number of nodes, fitting well the diffusion approximation, which is scalable and decomposes network analysis into individual nodes.

The numerical examples that we exhibit show that the transient regime can differ very significantly from the steady-state results. Since SDN controllers frequently change the state of the network paths, including the resulting traffic intensities and the load of each router, our numerical results show that the transient analysis will be indispensable in evaluating the performance of routers in a SDN network.

1.1 Contributions of this paper

In this paper, we develop the transient analysis of the diffusion approximation approach for small multi-node networks in which SDN controllers' decisions cause frequent changes of the paths of flows and hence of the traffic carried by different forwarders or routers. We compute the dynamics of queue length distributions, queueing delays, and loss probabilities as a function of changes in traffic intensity, based on solving a system of partial differential equations for the queue length distributions and the delays of several interconnected nodes as a function of time. The analysis we develop allows us to compute a network's transient behaviour, and the time it takes for a network to reach its new steady-state after the input traffic rates change. We can also compute the transient and steady-state packet loss probabilities in cases when they may be small, and hence very difficult to obtain via discrete event simulations.

We compute the diffusion transient state step-by-step in short time intervals with parameters which are specific to each of these intervals. Thus SDN routing traffic decisions can easily be reflected in successive changes to time-dependent and state-dependent diffusion parameters. Transient path delay averages can also be computed from transient node delay averages.

The use of these results are illustrated with two applications:

- In all networks, and in particular those carrying IoT traffic, short and intermittent packet sequences carrying measurement data need to be conveyed rapidly towards a destination. At the same time, when SDN controllers are

used, long traffic sequences at higher traffic rates may be re-allocated between paths for purposes of traffic balancing and may disrupt the QoS of the short sequences. The question then is to determine when the short sequences should be forwarded, e.g. just after a major long connection ends, or just when it begins. Intuitively speaking, one may wish to wait for the end of the longer and higher traffic sequence in order to obtain a better QoS for the short packet sequence. The transient analysis recommends that the short sequence be superposed on the longer higher rate traffic just as the latter begins, which is counterintuitive, as discussed in Section 4.1.

– The second example is given in Section 4.2, and shows how Service Level Agreements (SLA) are strongly affected by the use of precise transient analysis rather than steady-state analysis.

2 Transient Analysis

The diffusion approximation replaces the number of packets in a queueing system by the real-valued valued diffusion process $X(t) \in [0, N]$ where N is the maximum size of the queue. Following the approach in [8], resulting in equations (1), at the extremities of the interval $x = 0$ and $x = N$, two absorbing barriers are placed so that when $X(t)$ reaches a barrier, it stays there for a random time and jumps from $x = 0$ to $x = 1$ with intensity λ and from $x = N$ to $x = N - 1$ with intensity μ. The resulting diffusion equation is:

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x}$$
$$+ \lambda p_0(t)\delta(x - 1) + \mu p_N(t)\delta(x - N + 1),$$
$$\frac{dp p_N(t)0(t)}{dt} = \lim_{x \to 0} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)] - \lambda p_0(t),$$
$$\frac{dp_N(t)}{dt} = \lim_{x \to N} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)] - \mu p_N(t), \qquad (1)$$

where $\delta(x)$ is the Dirac delta function, and $f(x, t; x_0) = P[x \leq X(t) < x + dx \mid X(0) = x_0]$ of $X(t)$; $p_0(t)$ and $p_N(t)$ denote probabilities that at time t the process is in barriers at $x = 0$, $x = N$, respectively. The incremental changes of $X(t)$, $dX(t) = X(t + dt) - X(t)$ are normally distributed with mean βdt and variance αdt where β, α are coefficients of the diffusion equation, where $\beta = (\lambda - \mu)$ and $\alpha = (\sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3)$; $1/\lambda$ and $1/\mu$ are the mean interarrival and service times, and σ_A^2, σ_B^2 are the variances of the interarrival and service time, respectively.

To determine the solution of (1) we use the following appoach from [36]. First we consider a diffusion process with two *absorbing* barriers at $x = 0$ and $x = N$, started at $t = 0$ from $x = x_0$. Its probability density function $\phi(x, t; x_0)$ has the following form [37]:

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

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 $x_n' = 2nN$, $x_n'' = -2x_0 - x_n'$.

If the initial condition is defined by a function $\psi(x)$, $x \in (0, N)$, $\lim_{x \to 0} \psi(x) = \lim_{x \to N} \psi(x) = 0$, then the probability density function (pdf) of the process is $\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi)\psi(\xi)d\xi$.

The probability density function $f(x, t; \psi)$ of the diffusion process with elementary returns is composed of the function $\phi(x, t; \psi)$ referring to the diffusion process before it reaches any barrier, and of a spectrum of functions $\phi(x, t-\tau; 1)$, $\phi(x, t-\tau; N-1)$. The latter functions represent diffusion processes with absorbing barriers at $x = 0$ and $x = N$, started with densities $g_1(\tau)$ and $g_{N-1}(\tau)$ at time $\tau < t$ at points $x = 1$ and $x = N - 1$ due to jumps from the barriers:

$$f(x, t; \psi) = \phi(x, t; \psi) + \int_0^t g_1(\tau)\phi(x, t - \tau; 1)d\tau$$

$$+ \int_0^t g_{N-1}(\tau)\phi(x, t - \tau; N - 1)d\tau \tag{3}$$

where the densities $g_1(\tau)$, $g_{N-1}(\tau)$, as well as $p_0(t)$ and $p_N(t)$, are obtained from the probability balance equations at the barriers.

The delay through the queue, including waiting and service time, is then obtained as a first passage time from an initial point taken with probability given by $f(x, t; \psi)$ to the absorbing barrier placed at $x = 0$.

3 Transient Analysis of a Network

Consider a network of M stations with general service time distributions and routing probabilities r_{ij}. We first decompose the network by determining the input flows at each station and then apply the single server model to each station separately.

In the transient state the input flow $\lambda_{i-in}(t)$ of any station i and its output flow $\lambda_{i-out}(t) = (1 - p_0(t))\mu_i$ are different. The traffic equations balancing the flows of stations are

$$\lambda_{i-in}(t) = \lambda_{0i}(t) + \sum_{j=1}^M \lambda_{j-out}(t)r_{ji}, \ i = 1, \ldots, M, \tag{4}$$

where the first term λ_{0i} represents traffic flow coming directly to station i from the outside of the network. Denote by $f_{Aj}(x, t)$ and $f_{Bj}(x, t)$ the density functions of interarrival and service times at station j; the pdf $f_{Dj}(x, t)$ of the interdeparture times from this node at time t is

$$f_{Dj}(x, t) = \varrho_j(t)f_{Bj}(x, t) + [1 - \varrho_j(t)]f_{Aj}(x, t) * f_{Bj}(x, t), \ i = 1, \ldots, M \tag{5}$$

where * denotes the convolution and $i = 1, \ldots, M$. The first term of (5) represents the interdepature times of packets when the node is busy, and the second term gives the interdeparture times when it is idle. The formula (5), known as Burke's theorem, is exact for Poisson input and approximate in other cases. From (5) we have:

$$C_{Dj}^2(t) = \varrho_j^2(t)C_{Bj}^2(t) + C_{Aj}^2(t)(1 - \varrho_j(t)) + \varrho_j(t)[1 - \varrho_j(t)]. \qquad (6)$$

Packets leaving the node j according to the distribution $f_{Dj}(x, t)$ choose any node i with probability r_{ji} and the times between packets routed from node j to i has pdf:

$$f_{ji}(x, t) = f_{Dj}(x, t)r_{ji} + f_{Dj}(x, t) * f_{Dj}(x, t)(1 - r_{ji})r_{ji} + \cdots \qquad (7)$$

The variance of $f_{ji}(x, t)$ allows us to determine the variance of the number of customers going from station j to i, and after summing over all stations sending packets to station i we receive

$$C_{Ai}^2(t) = \frac{1}{\lambda_{i-in}(t)} \sum_{j=1}^{M} r_{ji}\lambda_{i-out}(t)[(C_{Dj}^2(t) - 1)r_{ji} + 1] + \frac{C_{0i}^2(t)\lambda_{0i}(t)}{\lambda_{i-in}(t)}, \quad (8)$$

where the parameters λ_{0i} and C_{0i}^2 refer to the flow coming to i from outside of the network, and (6), (8) form a system of linear equations yilding $C_{Ai}^2(t)$ and, in consequence, the diffusion parameters $\beta_i(t)$, $\alpha_i(t)$ for evry node i.

If $f_{Ri}(x, t)$ is the response time pdf at node i, then the response time pdf for the path $1, \ldots, n$ is $f_R = f_{R1}(x, t) * f_{R2}(x, t) * f_{R3}(x, t) * \cdots * f_{Rn}(x, t)$.

3.1 Packet Service Time for a SDN Data Plane Router

In SDN, routes are selected by one or more SDN controllers, while the SDN data plane routers are forwarding devices that follow the rules given by the controller. The centralisation of network intelligence and management in the controller enables a global network view, network programmability and deployment of innovative approaches such as smart cognitive routing [27].

Since the input and output hardware of an SDN forwarder is fast, the actual forwarding time between nodes can be neglected. However, the main component of service time that needs to be considered is the time during which the node's hardware identifies – for each successive packet – the flow to which the packet belongs, and in which output port the packet must be placed:

– Assume that the identification of the flow is conducted as a linear search in a flow table with K entries (i.e. the number of flows), and that T is the time to check one entry. If p is 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 which is a constant service time with zero variance.

– Otherwise, with probability $(1 - p)$, the time to find the existing entry is uniformly distributed in $[T, KT]$ for a simple linear search, since the packet is equally likely to belong to any of the flows, with mean $\frac{(K+1)T}{2}$ and variance $\frac{(K^2-1)T^2}{12}$.

– As a result, the service time S has the following mean and variance:

$$E[S] = T[pK + (1-p)\frac{K+1}{2}], \quad V(S) = (1-p)\frac{(K^2-1)T^2}{12}.$$
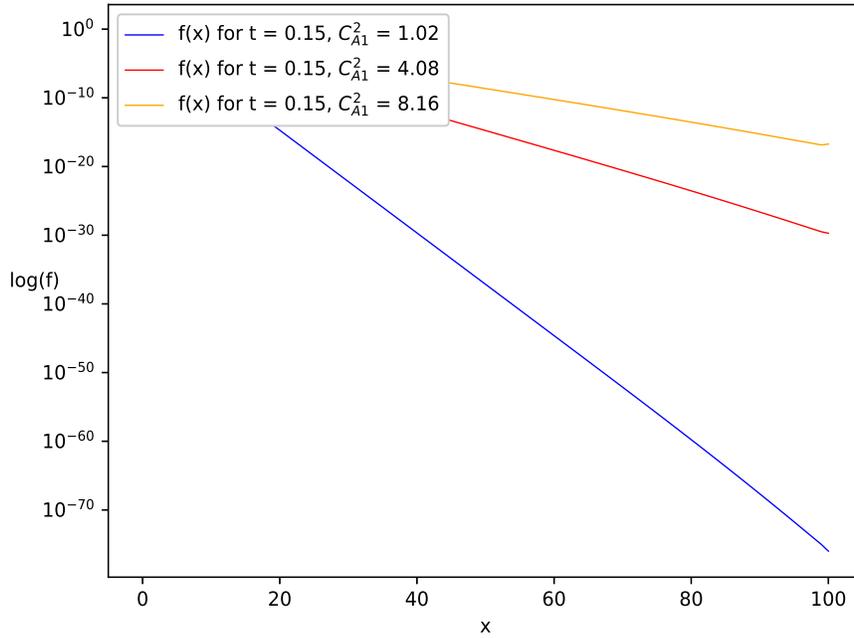
Fig. 1. Queue length Probability Density Function for node S1 at time $t = 0.15$ with different values of C_{A1}^2.

4 Numerical Examples

We consider a network composed of 4 forwarders. Host 1 is sending packets to Host 2 through forwarders S1-S2-S4 or S1-S3-S4, with routing probabilities $r_{12} = 0.15$, $r_{13} = 0.85$. The packet traffic rate from Host 1 is denoted $\lambda(t)$ in the range of 500 to 2500 packets/sec, and the changes in the traffic rate the pattern of is displayed in blue in all the figures. It is the total flow sent by S1; the flows
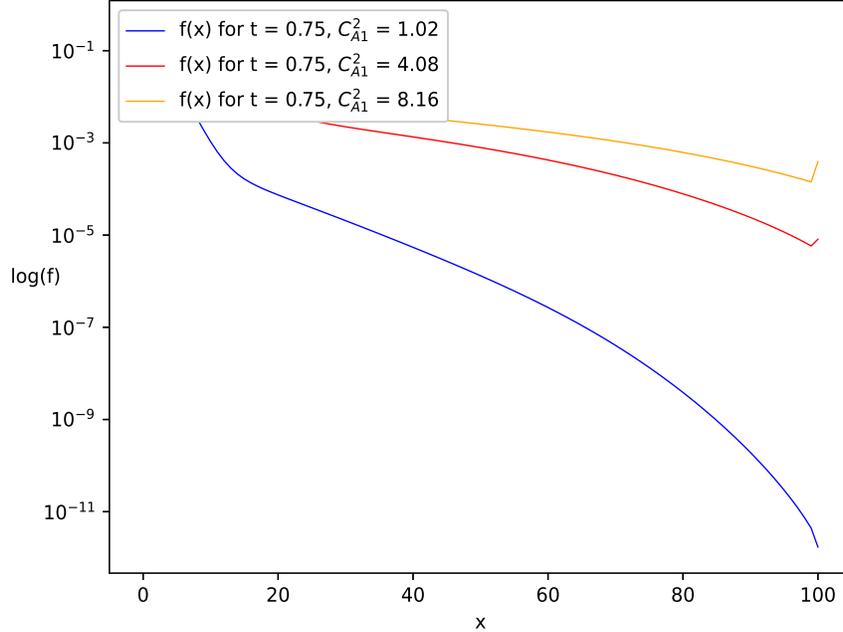
Fig. 2. Queue length pdf for node S1 at time $t = 0.75$ with different values of C_{A1}^2.

of S2 and S3 are defined by the routing probabilities. The duration of the time interval being considered is 1 second.

Traffic data from the CAIDA traces [38] concerning IPv4 packet interarrival times from the Equinix Chicago link, collected during one hour on 18 February 2016, gave $C_{A1}^2 = 1.02$ with over 22 million packets belonging to over 1:17 million IPv4 flows. The computations were also carried out for four and eight times larger, values C_{A1}^2 to see how these variations influence the network's performance. We assume that the switches store $K = 950$ flows and that the time to examine one flow in the flow table is $T = 8 \cdot 10^{-7}$sec with $p = 0$. The resulting mean service time is $S = 1/\mu = 0.038$msec or $\mu = 2,631.5$ packets/sec and $C_B^2 = 0.33$. The buffer capacity per flow is $N = 100$. These values are compatible with existing equipment but may vary with the type of router.

The transient solution is obtained numerically for 100 successive sub-intervals of the length 10 msec in 100 sub-intervals with fixed diffusion parameters in each sub-interval. At the end of each sub-interval (4), (8) are solved to determine new parameters of the flow for the single station models in the next interval. The density function $f_i(x)$ obtained for any station i at the end of an interval gives the initial conditions for the diffusion equation at the next interval.

Figure 1 and 2 present $f(x, t : \psi)$ given by (3) for station S1. In the first case, the buffer is relatively empty, i.e. the probability of the queue size being close to N is of the order of 10^{-30} or 10^{-70}. Note that the probability scale
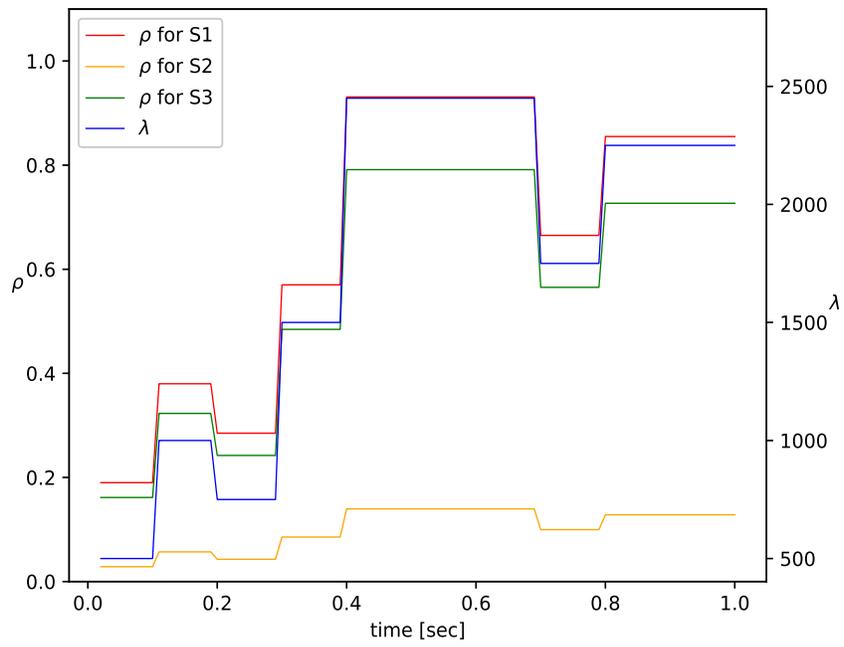
Fig. 3. The utilisation $\varrho(t) = 1 - p(0, t)$ and $C_{A1}^2 = 1.02$ for nodes S1, S2, S3, S4.
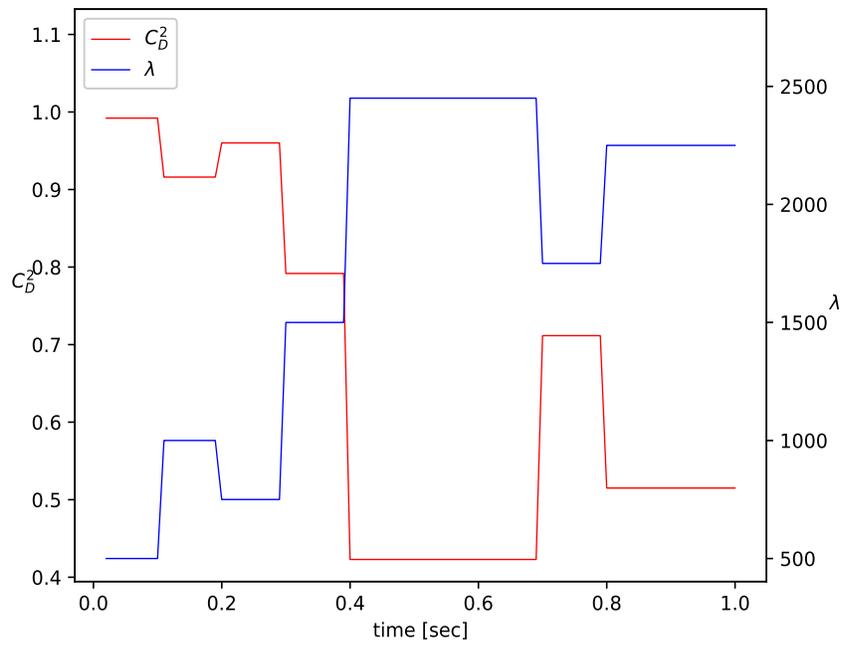
Fig. 4. $C_{D1}^2(t)$ for node S1.
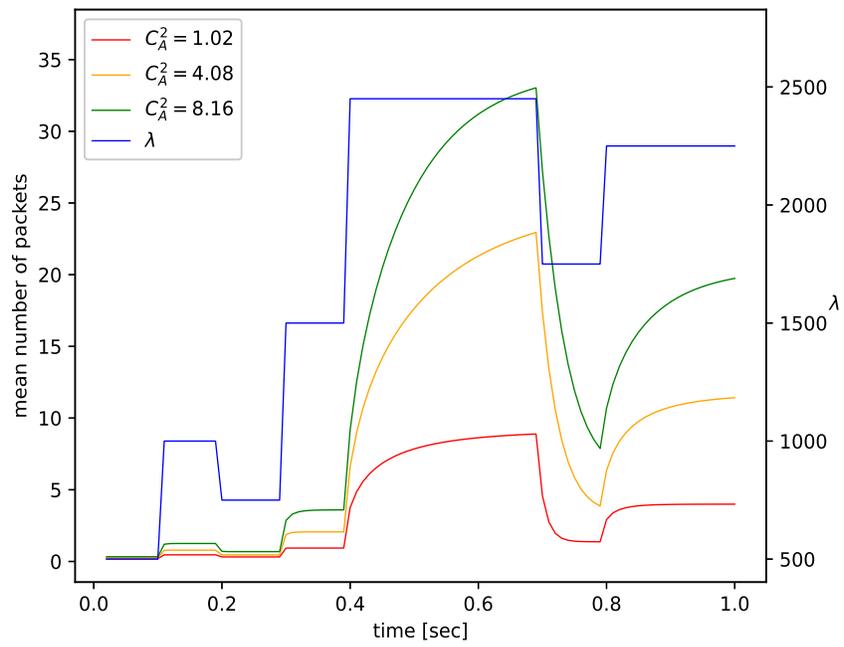
Fig. 5. Station S1, mean number of packets, for different values of C_{A1}^2.
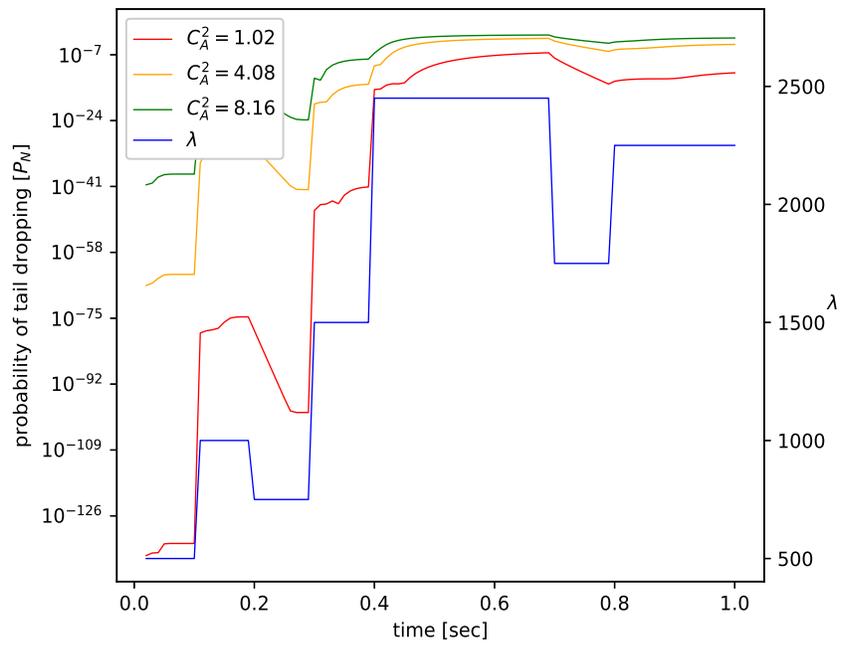
Fig. 6. Station S1, loss probability $p(N)$, for different values of C_{A1}^2.
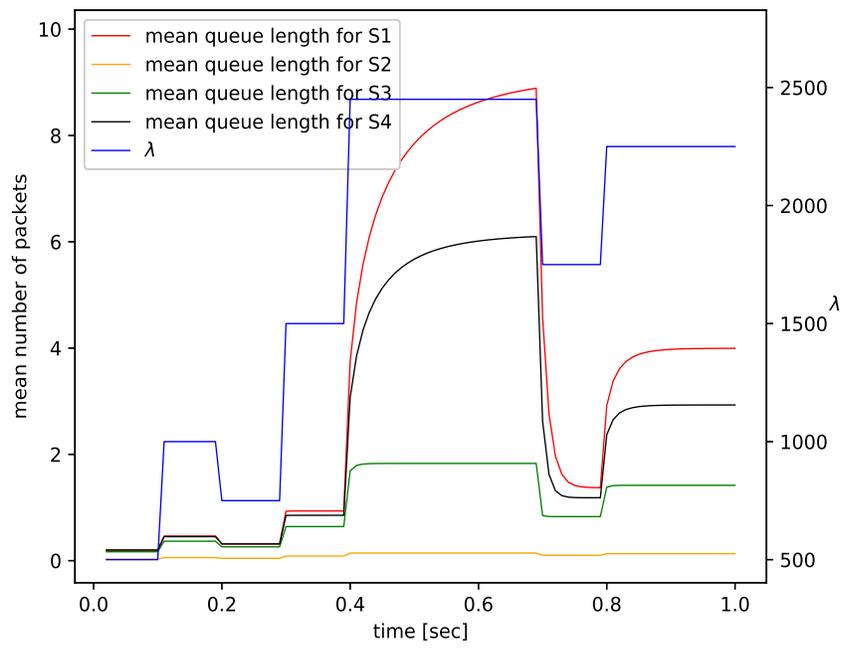
Fig. 7. Mean number of packets at S1, S2, S3, S4, for paths S1-S2-S4 and S1-S3-S4 paths.
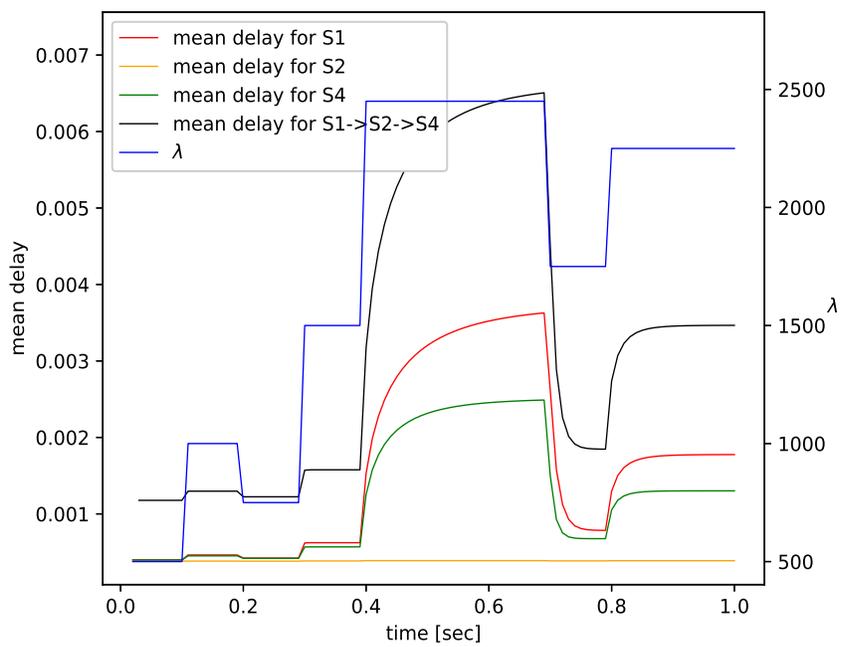
Fig. 8. Mean delays at S1, S3, S4 and total mean delay for the path S1-S2-S4, for $C_{A1}^2 = 1.02$
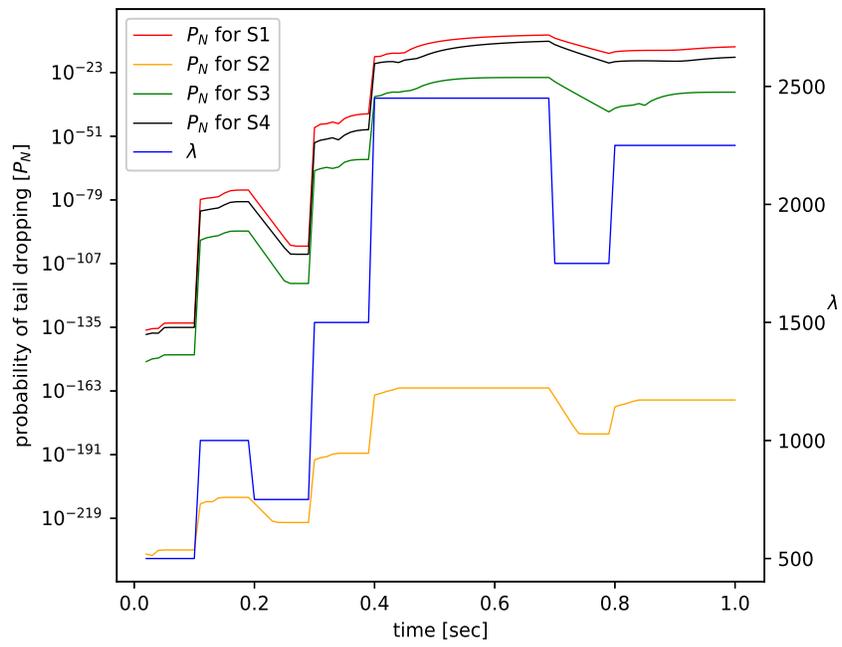
Fig. 9. Packet Loss Probability with $C_{A1}^2 = 1.02$, for nodes S1, S2, S3, S4, $p(N, t)$
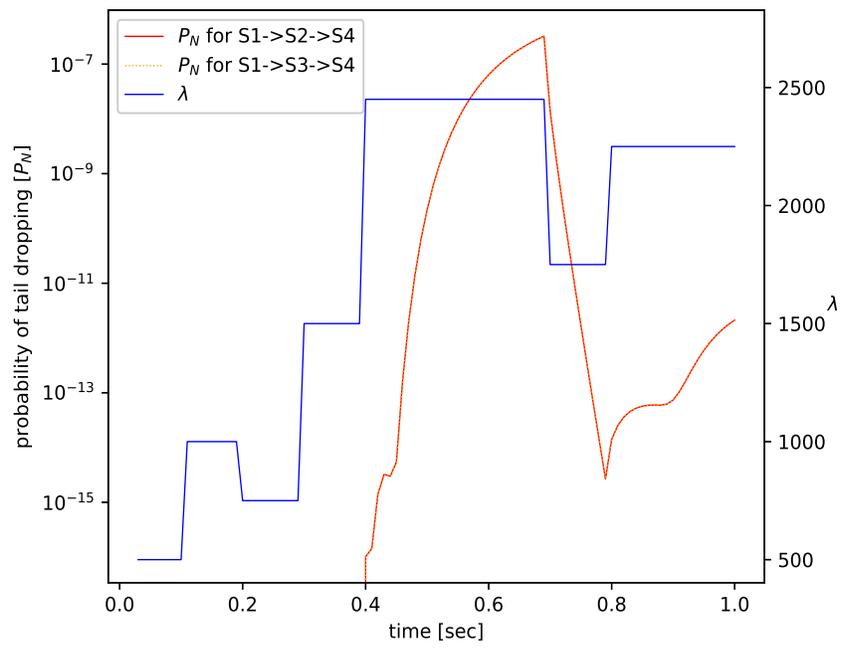
Fig. 10. Packet Loss Probability with $C_{A1}^2 = 1.02$, for paths S1-S2-S4 and S1-S3-S4 paths

is logarithmic, and the method has no difficulty to compute such small values, which would be impossible in a simulation model. We see also the impact of C_{A1}^2 which affects α in the diffusion equation; a greater C_{A1}^2 increases the probability of larger queue lengths. Figures 3 and 4 show the changes in the utilisation rate $\varrho_i(t)$ of stations and the interdeparture time C_{D1}^2 from (6), as $\lambda(t)$ is varied, showing that $\varrho(t)$ and C_{D1}^2 follow the variations in $\lambda(t)$ closely.

Figures 5 and 6 refer to station S1 and shows the variation of mean queue length for various values of C_{A1}^2. The queue length increases considerably with the variance of interarrival times, and the duration of the transient period also increases. Higher utilisation rates due to larger λ lead to a longer transient period. For large λ, the steady-state is not attained before the next change of the input flow rate. The right-hand figure illustrates the significant influence of C_{A1}^2 on the loss probabilities, which are presented in logarithmic scale.

Figures 7 and 8 present the dynamics of the average queue length and delay at stations S1, S2, S4, and the total average delay along the path S1-S2-S4. Again, we see the influence of the traffic rate on the queue length and the duration of the transient period. For higher traffic rates, the transient time is, in general, longer than the periods between changes in λ so that steady-state is never reached.

Figures 9 and 10 display the loss probabilities for each station separately, and the total loss probability for two paths S1-S2-S4, S1-S3-S4, which are practically the same and they superimpose. These figures also illustrate the method's ability to compute very small probabilities of the order of 10^{-200}.

4.1 Scheduling Short Sequences of Packets

Consider a short sequence of a few packets (of very low traffic rate) containing measurement traffic, emanating for instance from some IoT sensor, which is very sensitive to packet loss. Suppose the SDN controller establishes a connection, to be able to forward these packets at time 0.4 seconds along path $S1->S2->S4$. At the same time, the SDN controller also establishes a connection for another flow totalling 1000 packets/sec. Suppose also that when this connection was established this same path was already carrying 1500 packets/sec.

The question then is whether the source of the low data rate measurement traffic should wait for the high data rate traffic to end before forwarding its traffic. Of course, waiting has the disadvantage of incurring the waiting delay, which may be quite long. However, if the data traffic is very sensitive to losses, it may be better to wait for the high traffic rate flow to end.

The analysis that is illustrated in Figure 10 tells us that the low data rate IoT source should not wait at all. It should send its traffic *right away* as soon as the path $S1->S2->S4$ is established for the IoT traffic at time 0.4 seconds, simply because of the transient behaviour of the packet loss probability which is less than 10^{-13} in the period just after time 0.4 seconds. In addition to avoiding the wait for the high traffic rate to end, the instant when the high traffic rate ends at time 0.7 seconds and the following 0.1 seconds will have a much higher packet loss probability of the order of 10^{-7} due to the transient effect in packet loss. In addition, the transient analysis of Figure 8 also tells us that the mean

wait delay incurred by the packets is of the order of 0.004 seconds, rather than the 0.3 seconds that would be wasted waiting for the high traffic rate to subside.

Obviously, this type of very useful insight about the transients in packet loss and delay can only be obtained via the transient analysis tools developed in this paper.

4.2   The Effect of Transients on Service Level Agreements

A high priority customer of the network needs to send a flow of packets every 30 minutes from node 1 to node 4 during a short time window of $\Delta = 200$ milliseconds. The customer indicates that the traffic rate λ_1 may take a fixed value between 2500 and 3200 packets/second, and that that the inter-arrival time distribution for packets will have a squared coefficient of variation which does not exceed $C = 2$.

- The customer has stringent QoS constraints so that the network operator must abide by a Service Level Agreement (SLA): the total average packet delay through the path must not exceed $W_m = 0.01$ during the connection, and the total average packet loss over the path must not exceed $L_m = 150$.
- Due to the existing network topology, to reach node 4, the traffic must travel through the three identical router s$S1-> S2-> S4$. The network operator will program a SDN controller to set up a private 3-node path $S1-> S2-> S4$ for this flow, at the beginning of each successive 30 minute interval, and reserve it (empty of other traffic) for 200 milliseconds. After 100 milliseconds, this flow stops sending packets, and the SDN controller can re-allocate the path to other flows.
- The operators current routers have a 2000 packet/second forwarding capacity. The operator would like to know if this is sufficient, or whether she/he should upgrade the hardware to higher available speeds of 2300, or 2600 or 2900 packet/second.

Let $t = 0$ be the beginning of an interval of length Δ, and $N_i(t)$ be the average number of packets from the flow in router i, at t seconds after the beginning of the flow. Let:

$$W_{max} = S \max_{t \in \Delta}[N_1(t) + N_2(t) + N_4(t)], \tag{9}$$

$$L_{max} = \max_{t \in \Delta}[L_1(t) + L_2(t) + L_4(t)], \tag{10}$$

where W_{max}, L_{max} are the worst case values of the packet delay and loss for the flow rate, starting at $t \in [0, \Delta]$ Due to the existing network topology, to reach Node 4 the traffic must travel through the three identical routers $S1-> S2-> S4$. After $\Delta = 200$ milliseconds this particular flow stops sending packets, and the SDN controller can re-allocate these nodes to other flows. We compute $L_i(t) = \lambda_i Prob[f_i(N, t)]$, where N is the router buffer size, and obtain W_{max}, L_{max} as a function of λ_1 for the worst case value $C_{A1}^2 = 2$, and different router speeds, to see which speed is needed.

The transient diffusion model yields W_{max}, L_{max} as shown in Figures 11 and 12, (in full lines) plotted against λ_1 for different values of $S = \mu^{-1}$. The figures also show (in dotted lines) the values of the steady-state values of average delay and loss rate.

$$W = \frac{1}{\mu} \lim_{t \to \infty} [N_1(t) + N_2(t) + N_4(t)], \tag{11}$$

$$L_{max} = \lim_{t \to \Delta} [L_1(t) + L_2(t) + L_4(t)], \tag{12}$$

We see that the transient analysis allows us to operate safely with routers having a capacity of $\mu = 2000 packets/sec$, while the steady-state analysis over-estimates the average delay and loss probability by over 100%, and recommends upgrading the routers to a packet processing capacity of at least 2600 packets/sec.
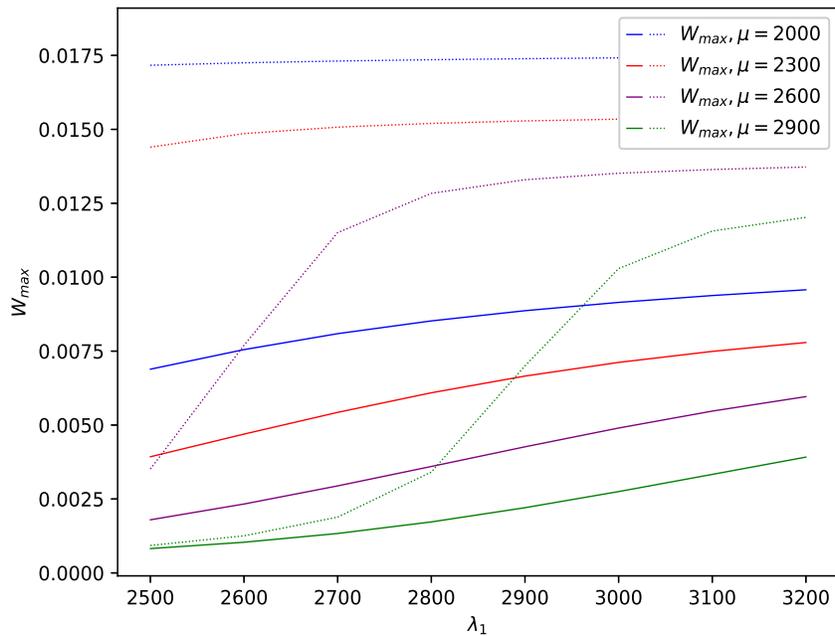
Fig. 11. W_{max} as in Eq. (12) for various λ_1 (solid line) compared with steady state solution (dotted line)

5   Conclusions

This paper considers a small network of routers or forwarders in a network controlled by a SDN controller which makes changes to paths through the network,
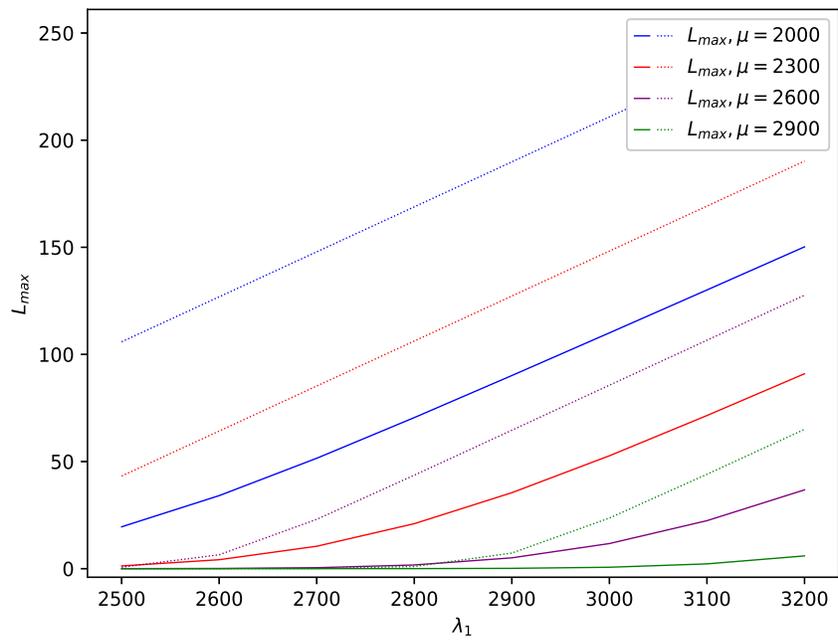
Fig. 12. L_{max} as in Eq. (12) for various λ_1 (solid line) compared with steady state solution (dotted line)

and regularly modifies traffic rates at different routers. Our purpose is to evaluate the effect of transient effects and in particular their importance relative to usual steady-state analyses which have been studied. We, therefore, develop a computationally efficient diffusion approximation method, present its analytical solution, and implement the numerical techniques needed for the transient computations. Using realistic parameters, including relatively low-frequency changes of paths by SDN controllers every 100ms, we show that the system may seldom reach steady-state when the network is moderately to heavily loaded which is the region of interest for performance modelling and optimisation studies. Though at light loads transients are short, at moderate to heavy loads transients are significant for individual node and path delays, and packet loss probabilities. The method we have developed is operational and gives quantitative results for models with realistic parameters.

Numerical examples for single and multiple node models provide the dynamics of queue lengths, delays and packet loss and their dynamics in response to changes in the flow intensity and the variance of interarrival times. Through numerical examples, we also show how transient analysis provides insights that are more accurate and cover cases where the steady-state analysis would provide wrong or incomplete results. Thus we conclude that transient analysis can play a major role in the performance evaluation of SDN networks and should be incorporated into SDN controls and that diffusion approximations can be useful and computationally efficient for this purpose.

6   Acknowledgements

References

1. L. Kleinrock, *Queueing Systems Vol. 1: Theory.*   John Wiley & Sons, 1975.
2. ——, *Queueing Systems Vol. 2: Computer Applications.* John Wiley & Sons, 1976.
3. H. Toral-Cruz, A. sakib Khan Pathan, and J. C. R. Pacheco, "Accurate modeling of voip traffic qos parameters in current and future networks with multifractal and markov models," *Mathematical and Computer Modelling*, vol. 57, no. 11–12, pp. 2832–2845, 2013.
4. R. Buyya et al., "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–38, 2019.
5. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

6. D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive resource management and control in software defined networks," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 1, p. 1833, March 2015.

7. W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

8. E. Gelenbe, "On approximate computer systems models," *Journal of the ACM*, vol. 22, no. 2, pp. 261–269, 1975.

9. T. G. Kurtz, "Limit theorems for sequences of jump markov processes approximating ordinary differential processes," *Journal of Applied Probability*, vol. 8, no. 2, p. 344356, 1971.

10. M. Fidler, "Survey of deterministic and stochastic service curve models in the network calculus," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, pp. 59–86, 2010.

11. M. Fidler and A. Rizk, "A guide to the stochastic network calculus," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 92–105, 2015.

12. K. Mahmood, A. Chilwan, O. Osterbo, and M. Jarschel, "Modelling of openflow-based software-defined networks: the multiple node case," *The Institution of Engineering and Technology Journal*, vol. 4, no. 5, pp. 278–284, 2015.

13. J. Ansell, W. K. G. Seah, B. Ng, and S. Marshall, "Making queueing theory more palatable to sdn/openflow-based network practitioners," in *Proceeding of the 2016 IEEE/IFIP Network Operations and Management Symposium*. Istanbul, Turkey: IEEE, 2016, pp. 1119–1124.

14. D. Singh, B. Ng, Y.-C. Lai, Y.-D. Lin, and W. K. G. Seah, "Modelling software-defined networking: Software and hardware switches," *Journal of Computer Network and Computer Applications*, vol. 122, pp. 24–36, 2018.

15. Y.-C. Lai, A. Ali, M. Hassan, S. Hossain, and Y.-D. Lin, "Performance modeling and analysis of tcp connections over software defined networks," in *Proceeding of the 2017 IEEE Global Communications Conference*. Singapore: IEEE, 2017, pp. 1–6.

16. A. Fahmin, Y.-C. Lai, S. Hossain, Y.-D. Lin, and D. Saha, "Performance modeling of sdn with nfv under or aside the controller," in *Proceeding of the 5th International Conference on Future Internet of Things and Cloud Workshops*. Prague: IEEE, 2017, pp. 211–216.

17. W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software defined networking under bursty multimedia traffic," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 55, pp. 24–36, 2018.

18. S. Azodolmolky, P. Wieder, and R. Yahyapour, "Performance evaluation of a scalable software-defined networking deployment," in *Proceeding of the 2013 Second European Workshop on Software Defined Networks*. Berlin, Germany: IEEE, 2013, pp. 68–74.

19. S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *Proceeding of the IEEE Global Communications Conference*. Atlanta, USA: IEEE, 2013, pp. 1397–1402.

20. T. Czachórski, E. Gelenbe, G. S. Kuaban, and D. Marek, "Transient behaviour of a network router, accepted," in *Proceedings of International Conference on Telecommunications and Signal Processing TSP 2020*, N. Herencsár, F. Benedetto, and J. Hosek, Eds. Milano: IEEE, 2020.

21. T. Czachórski and F. Pekergin, "Diffusion approximation as a modelling tool," in *Network Performance Engineering*, D. D. Kouvatsos, Ed., vol. 5233. Springer, 2011, pp. 447–476.

22. S. K. Lee, M. Bae, and H. Kim, "Future of iot networks: A survey," *Applied Sciences*, vol. 7, no. 10, 2017.

23. E. Gelenbe, J. Domanska, P. Frohlich, M. Nowak, and S. Nowak, "Self-aware networks that optimize security, qos and energy," *Proceedings of the IEEE, accepted for publication*, vol. 108, no. 7, 2020.

24. E. Gelenbe, "Energy packet networks: adaptive energy management for the cloud," in *CloudCP'12: Proceedings of the 2nd International Workshop on Cloud Computing Platforms*. ACM, 2012, pp. 1–5. [Online]. Available: https://doi.org/10.1145/2168697.2168698

25. S. Dobson et al., "A survey of autonomic communications," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 2, pp. 223–259, 2006.

26. E. Gelenbe, P. Liu, and J. Lainé, "Genetic algorithms for route discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 6, pp. 1247–1254, 2006.

27. F. Francois and E. Gelenbe, "Towards a cognitive routing engine for software defined networks," in *Proceeding of the 2016 IEEE International Conference on Communications*. Kuala Lumpur: IEEE, 2016, pp. 1–6.

28. Y. Goto, B. Ng, W. K.G.Seah, and Y. Takahashi, "Queueing analysis of software defined network with realistic openflow-based switch model," *Computer Networks*, vol. 164, 2019.

29. K. Sood, S. Yi, and Y. X. and, "Performance analysis of software-defined network router using m/geo/1," *IEEE communication Letters*, vol. 20, no. 12, pp. 27–51, 2016.

30. Z. Bozakov and A. Rizk, "Aming sdn controllers in heterogeneous hardware environments," in *Proceeding of the 2013 Second European Workshop on Software Defined Networks*. Berlin, Germany: IEEE, 2013, pp. 50–55.

31. N. Bisnik and A. A. Abouzeid, "Queuing network models for delay analysis of multihop wireless ad hoc networks," *Ad Hoc Networks 7*, vol. 7, no. 1, pp. 79–97, 2009.

32. E. Gelenbe and G. Pujolle, "The behaviour of a single-queue in a general queueing network," *Acta Inf.*, vol. 7, pp. 123–136, 1976. [Online]. Available: https://doi.org/10.1007/BF00265766

33. E. Gelenbe, "Probabilistic models of computer systems part ii: Diffusion approximations, waiting times and batch arrivals," *Acta Informatica*, vol. 12, pp. 285–303, 1979. [Online]. Available: https://doi.org/10.1007/BF00268317

34. G. A. Marin, X. Mang, E. Gelenbe, and R. O. Önvural, "Statistical call admission control," *US Patent 6,222,824*, 2001.

35. A. Domański, J. Domańska, T. Czachórski, J. Klamka, J. Szygula, and D. Marek, "Diffusion approximation model of tcp newreno congestion control mechanism," *SN COMPUT. SCI.*, vol. 1,43, 2020.

36. T. Czachórski, "A method to solve diffusion equation with instantaneous return processes acting as boundary conditions," *Bulletin of Polish Academy of Sciences, Technical Sciences*, vol. 41, no. 4, 1993.

37. R. P. Cox and H. D. Miller, *The Theory of Stochastic Processes*. London, UK: Chapman and Hall, 1965.

38. [Online]. Available: https://data.caida.org/datasets/passive-2016/equinix-chicago/20160218-130000.UTC/