Time Dependent Diffusion Model for Security Driven Software Defined Networks

Tadeusz Czachórski[1], Erol Gelenbe[1], Godlove Suila Kuaban[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. We present a model of a Software Defined Network (SDN)
where frequent changes in routing and traffic rates at routers are needed
to respond to the security, quality of service (QoS), and energy savings re-
quirements of applications such as the Internet of Things. Such frequent
path and traffic changes introduce time-dependent network behaviours,
and standard queueing models are not well adapted to analyse the tran-
sient regime, we propose a tractable diffusion approximation for both the
transient and steady-state behaviour. Our model can represent any net-
work topology transmitting time-dependent flows with routing changes,
and computes queue length and delay distributions at each network node
and along complete paths between senders and receivers. Using realistic
router parameters, we show that transients occupy a significant fraction
of system time, so that the optimisation conducted with SDN controllers
needs to include the effect of time-dependent behaviours.

Keywords: SDN, IoT Networks, Security, QoS, Routing, Transients,
Diffusion Approximation

1   Introduction

The Internet of Things (IoT) and its increasing volumes of traffic for new ser-
vices such as video related to security, server virtualisation of the Cloud and
Fog [38,8] and highly distributed data storage [22,9], create new challenges for
the Internet [26,35]. Indeed, expanding IoT applications such as Health Moni-
toring [32], Smart Homes [3] and Smart Vehicles [18], create large volumes of
intermittent traffic with stringent security, QoS and energy minimisation needs
[6].

Thus network structures based on static switches are not well suited to deliver
high performance, energy efficiency and reliability in such dynamically chang-
ing environments, and are not flexible enough to maintain Quality of Service
(QoS) for increasingly complex networks. On the other hand, SDN [34,39] with
intelligent programmable controllers can be aware of the overall state of nodes

and links, and dynamically manage the network and adapt to new conditions [25]. Indeed, SDN provides flexible and scalable routing for intelligent networks [14] by separating the control and data planes for traffic engineering, link failure recovery, load balancing [40] and security issues [41]. Thus the concentration of network intelligence and management in SDN controllers enables innovative smart cognitive routing [17,21] to respond by changing network paths and traffic levels to meet the dynamic security, QoS and energy savings requirements of the IoT.

Earlier studies of SDN switches have used steady-state queueing models such as $M/M/1$, $M/H_2/1$, $M/G/1$, $M/Geo/1$, $GI/M/1/K$, based on Markov chains, embedded Markov chains [29,2,36,28,16,31] or network calculus [4,5]. Thus they do not consider the frequent traffic changes due to controller decisions. To address this concern, we recently considered a single SDN forwarder and modelled it with a diffusion approximation [13], and considered a network of forwarders [12] to determine its transient behaviour. These studies have shown that under certain conditions, the transient regime can become dominant so that SDN based optimisation should consider the effect of transients.

In SDN, paths are selected by a controller, and the SDN data plane routers are then simple forwarding devices that follow the rules given by the controller. An analysis of the performance of SDN switches and their cooperation with the controller may be found in [33,27,40].

Therefore this paper, we extend these studies to address a SDN based network that supports IoT applications, and modifies its paths and traffic levels to respond to unpredictable changes in security and QoS, so that the network has time-dependent routing. To address this challenge we apply a diffusion approximation [24,20,30] which is well suited to investigate transient queueing problems with general interarrival and service time distributions for realistic network data.

The next section details the method for a single network node, while the mathematical model of time-dependent routing in the network is presented in Section 3 where the system equations such as (12), ... , (16) include routing probabilities which are functions of time, leading to a novel approach in diffusion models. Numerical examples are provided in Section 4 and conclusions are drawn in Section 5.

2　Single Node Transient Analysis

The diffusion approximation replaces the number of packets in a queueing system by the real-valued diffusion process $\{X(t)\} \in [0, N]$ where N is the maximum size of the queue. Following the approach in [19,23], at the extremities $x = 0$ and $x = N$ of the diffusion interval, 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:

$$\frac{\partial f(x,t;x_0)}{\partial t} = \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta\frac{\partial f(x,t;x_0)}{\partial x} +$$
$$+\lambda p_0(t)\delta(x-1) + \mu p_N(t)\delta(x-N+1)\,,$$
$$\frac{dp_0(t)}{dt} = \lim_{x\to 0}\left[\frac{\alpha}{2}\frac{\partial f(x,t;x_0)}{\partial x} - \beta f(x,t;x_0)\right] - \lambda p_0(t)\,,$$
$$\frac{dp_N(t)}{dt} = \lim_{x\to N}\left[-\frac{\alpha}{2}\frac{\partial f(x,t;x_0)}{\partial x} + \beta f(x,t;x_0)\right] - \mu p_N(t)\,, \qquad (1)$$

where $\delta(x)$ is the Dirac delta function, $p_0(t)$, $p_N(t)$ are probabilities that the process is at the barrires at $x = 0$ or $x = N$, respectively, and $f(x,t;x_0)$ is probability density function (pdf) of the process $\{X(t)\}$

$$f(x,t;x_0)dx = P[x \le X(t) < x + dx \mid X(0) = x_0].$$

The incremental changes of $\{X(t)\}$, $dX(t) = X(t+dt) - X(t)$ are normally distributed with the mean βdt and variance αdt where β, α are coefficients of the diffusion equation. The changes of the process $\{N(t)\}$ during an interval θ tend to normal distribution with mean $(\lambda - \mu)\theta$ and variance $(\sigma_A^2\lambda^3 + \sigma_B^2\mu^3)\theta$ where $1/\lambda$ 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 times, respectively. The choice

$$\beta = \lambda - \mu \quad \text{and} \quad \alpha = \sigma_A^2\lambda^3 + \sigma_B^2\mu^3 = C_A^2\lambda + C_B^2\mu,$$

where C_A^2, C_B^2 are squared coefficients of variation of interarrival and service times, assures that the changes of both processes $\{X(t)\}$ and $\{N(t)\}$ have normal distributions with the same parameters.

To determine the solution of (1) we use the following appoach from [11]. 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 [10]:

$$\phi(x,t;x_0) = \begin{cases} \delta(x-x_0) & \text{for } t = 0 \\ \frac{1}{\sqrt{2\Pi\alpha t}}\sum_{n=-\infty}^{\infty}\left\{\exp\left[\frac{\beta x_n'}{\alpha} - \frac{(x-x_0-x_n'-\beta t)^2}{2\alpha t}\right]\right. \\ \left. - \exp\left[\frac{\beta x_n''}{\alpha} - \frac{(x-x_0-x_n''-\beta t)^2}{2\alpha t}\right]\right\} & \text{for } t > 0\,, \end{cases}$$

(2)

where $x_n' = 2nN$, $x_n'' = -2x_0 - x_n'$.
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 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 jumps from the boundaries 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

$\phi(x, t - \tau; 1)$, $\phi(x, t - \tau; N - 1)$ representing 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:

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

First, we compute densities $\gamma_0(t)$, $\gamma_N(t)$ of probability that at time t the process enters to $x = 0$ or $x = N$ are

$$\gamma_0(t) = p_0(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,0}(t) + \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau)d\tau$$

$$+ \int_0^t g_{N-1}(\tau)\gamma_{N-1,0}(t - \tau)d\tau ,$$

$$\gamma_N(t) = p_N(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,N}(t) + \int_0^t g_1(\tau)\gamma_{1,N}(t - \tau)d\tau$$

$$+ \int_0^t g_{N-1}(\tau)\gamma_{N-1,N}(t - \tau)d\tau ,$$
(4)

where $\gamma_{1,0}(t)$, $\gamma_{1,N}(t)$, $\gamma_{N-1,0}(t)$, $\gamma_{N-1,N}(t)$ are densities of the first passage time between corresponding points, e.g.

$$\gamma_{1,0}(t) = \lim_{x \to 0}[\frac{\alpha}{2}\frac{\partial\phi(x, t; 1)}{\partial x} - \beta\phi(x, t; 1)] .$$
(5)

For absorbing barriers

$$\lim_{x \to 0}\phi(x, t; x_0) = \lim_{x \to N}\phi(x, t; x_0) = 0 ,$$

hence $\gamma_{1,0}(t) = \lim_{x \to 0} \frac{\alpha}{2}\frac{\partial\phi(x,t;1)}{\partial x}$. The functions $\gamma_{\psi,0}(t)$, $\gamma_{\psi,N}(t)$ denote densities of probabilities that the initial process, started at $t = 0$ at the point ξ with density $\psi(\xi)$ will end at time t by entering respectively $x = 0$ or $x = N$.

Finally, we may express $g_1(t)$ and $g_N(t)$ with the use of functions $\gamma_0(t)$ and $\gamma_N(t)$:

$$g_1(\tau) = \int_0^\tau \gamma_0(t)l_0(\tau - t)dt , \qquad g_{N-1}(\tau) = \int_0^\tau \gamma_N(t)l_N(\tau - t)dt , \quad (6)$$

where $l_0(x)$, $l_N(x)$ are the densities of sojourn times in $x = 0$ and $x = N$; the distributions of these times are not restricted to exponential ones as it is in Eq. (1).

Technicaly, it is easier to compute this solution in Laplace domain where convolutions of functions become products. For any function $h(t)$ we denote by $\bar{h}(s)$ its Laplace transform. The Laplace transform $\bar{f}(x, s; \psi)$ of the density function $f(x, t; \psi)$ is

$$\bar{f}(x, s; \psi) = \bar{\phi}(x, s; \psi) + \bar{g}_1(s)\, \bar{\phi}(x, s; 1) + \bar{g}_{N-1}(s)\, \bar{\phi}(x, s; N-1), \qquad (7)$$

and the Laplace transform of $\phi(x, t; x_0)$ can be expressed as

$$\bar{\phi}(x, s; x_0) = \frac{\exp[\frac{\beta(x - x_0)}{\alpha}]}{A(s)} \sum_{n=-\infty}^{\infty} \left\{ \exp\left[-\frac{|x - x_0 - x_n'|}{\alpha} A(s) \right] \right.$$
$$\left. - \exp\left[-\frac{|x - x_0 - x_n''|}{\alpha} A(s) \right] \right\}, \qquad (8)$$

where $A(s) = \sqrt{\beta^2 + 2\alpha s}$. For computational efficiency, we rearranged the Eq. (8) to the form

$$\bar{\phi}(x, s; x_0) = \frac{\exp[\frac{\beta(x - x_0)}{\alpha}]}{A(s)} \left\{ \mathbf{1}_{(x \geq x_0)} \left[\exp\left(-\frac{x A(s)}{\alpha} \right) 2\sinh\left(\frac{x_0 A(s)}{\alpha} \right) \right] \right.$$
$$+ \mathbf{1}_{(x_0 < x)} \left[\exp\left(-\frac{x_0 A(s)}{\alpha} \right) 2\sinh\left(\frac{x A(s)}{\alpha} \right) \right]$$
$$\left. - 2\sinh\left(\frac{x A(s)}{\alpha} \right) 2\sinh\left(\frac{x_0 A(s)}{\alpha} \right) \times \sum_{n=1}^{\infty} \exp\left(-2nN \frac{A(s)}{\alpha} \right) \right\}.$$

Similarily, $\bar{\phi}(x, s; \psi) = \int_0^N \bar{\phi}(x, s; \xi)\psi(\xi)d\xi$.

Laplace transforms of Eqs. (4), (6) give us $\bar{g}_1(s)$ and $\bar{g}_{N-1}(s)$:

$$\bar{g}_1(s) = \left\{ p_0(0) + \bar{\gamma}_{\psi,0}(s) + [p_N(0) + \bar{\gamma}_{\psi,N}(s)] \frac{\bar{l}_N(s)\bar{\gamma}_{N-1,0}(s)}{1 - \bar{l}_N(s)\bar{\gamma}_{N-1,N}(s)} \right\} \cdot$$
$$\cdot \frac{\bar{l}_0(s)}{1 - \bar{l}_0(s)\bar{\gamma}_{1,0}(s)} \left[1 - \frac{\bar{l}_0(s)\bar{\gamma}_{1,N}(s)}{1 - \bar{l}_0(s)\bar{\gamma}_{1,0}(s)} \frac{\bar{l}_N(s)\bar{\gamma}_{N-1,0}(s)}{1 - \bar{l}_N(s)\bar{\gamma}_{N-1,N}(s)} \right]^{-1},$$

$$\bar{g}_{N-1}(s) = \frac{\bar{l}_N(s)}{1 - \bar{l}_N(s)\bar{\gamma}_{N-1,N}(s)} \left[p_N(0) + \bar{\gamma}_{\psi,N}(s) + \bar{g}_1(s)\bar{\gamma}_{1,N}(s) \right].$$

Probabilities that at the moment t the process has the value $x = 0$ or $x = N$ are

$$\bar{p}_0(s) = \frac{1}{s} \left[\bar{\gamma}_0(s) - \bar{g}_1(s) \right], \qquad \bar{p}_N(s) = \frac{1}{s} \left[\bar{\gamma}_N(s) - \bar{g}_{N-1}(s) \right]. \qquad (9)$$

The inverse transforms of (7), (9) are obtained with the use of Stehfest's algorithm [37]. In this algorithm a function $f(t)$ is obtained from its transform $\bar{f}(s)$ for any fixed argument t as

$$f(t) = \frac{\ln 2}{2} \sum_{i=1}^{H} V_i\, \bar{f}\left(\frac{\ln 2}{t} i \right), \qquad (10)$$

where

$$V_i = (-1)^{H/2+i} \sum_{k=\lfloor \frac{i+1}{2} \rfloor}^{\min(i,H/2)} \frac{k^{H/2+1}(2k)!}{(H/2-k)!k!(k-1)!(i-k)!(2k-i)!}. \qquad (11)$$

H is an even integer; we used $H = 16$, following Stehfest's recommendations. Theoretically (10) is an infinite series and the increase of H should increase the accuracy of computations. However, considering the form of the functions being inverted in our case, the values of elements of (10) become either very small or very large with the increase of the index i and introduce numerical errors. After some numerical experiments, we found $H = 16$ is satisfactory and we do not need to introduce longer computer words for extra numerical precision.

Note that the presented transient solution is valid for constant diffusion parameters. However, the values of flows, hence also model parameters, may vary with time. Therefore in computations, we fix model parameters during small intervals δ (of the order of a single mean service time) and the solution at the end of one interval determines the initial conditions (i.e. function ψ in Eqs. (3), (7)) for the next interval.

The delay through the queue, including waiting and service time, is obtained as a first passage time from an initial point taken with probability density $f(x, t; \psi)$ to the absorbing barrier placed at $x = 0$, see [13].

It is known that $\lim_{s \to 0} s\bar{f}(s) = \lim_{t \to \infty} f(t)$ if $s\bar{f}(s)$ is an analytic function for $\Re(s) \geq 0$, therefore the above solution in form of Laplace transforms is convergent to steady-state solution for real domain.

3 Network of Nodes, Transient Analysis

Consider a network of M stations type G/G/1/N with routing probabilities $r_{ij}(t)$. We follow the approach of [24] developed for the steady-state network model, then adapted to transient analysis in [15]. Here we introduce additionally, for the needs of SDN, the time-depending routing.

The first objective of the network model is to decompose the network: to determine the input flows at every station and then apply the single server model of the previous section to each station separately.

In the transient state, we should distinguish at any station i the input flow $\lambda_{i-in}(t)$ and the output flow $\lambda_{i-out}(t)$

$$\lambda_{i-out}(t) = [1 - p_{0i}(t)]\mu_i,$$

which are different. $p_{0i}(t)$ denotes probability that the station i is idle at time t, i.e. the diffusion process related to this station is inside the barrier at $x = 0$. The term $1 - p_{0i}(t) = \varrho_i$ presents probability that the station i is busy and customers are leaving it with the rate μ_i.

The traffic equations balancing the flows of stations are

$$\lambda_{i-in}(t) = \lambda_{0i}(t) + \sum_{j=1}^{M} \lambda_{j-out}(t)r_{ji}(t), \qquad i = 1, \ldots, M, \qquad (12)$$

where the first term λ_{0i} represents traffic flow coming from the outside of the network directly to station i.

As mentioned earlier, routing probabilities $r_{ji}(t)$ are changing each interval Δ following decisions of the controler, remaining constant inside the interval, and flow parameters may change every interval $\delta < \Delta$; we assume for simplicity $\Delta = n\delta$, in numerical examples below $n = 10$. This way all model parameters are constant witin intervals δ when the solution (3) is computed.

Denote by $f_{Aj}(x,t)$ and $f_{Bj}(x,t)$ the density functions of interarrival and service times distributions at station j. The pdf $f_{Dj}(x,t)$ of the interdeparture times from this node at time t may be expressed as

$$f_{Dj}(x,t) = \varrho_j(t)f_{Bj}(x,t) + [1-\varrho_j(t)]f_{Aj}(x,t)*f_{Bj}(x,t) , \quad j = 1,\ldots,M, \quad (13)$$

where * denotes the convolution. The first term of the right side in (13) represents the interdepature times of packets when the node j is working and the second term gives the interdeparture times when it is idle. The formula (13), known as Burke's theorem [7], is exact for Poisson input (the pdf of the idle period distribution that should be used in the second term of (13) is the same as $f_{Aj}(x,t)$) and approximate in other cases. From (13) we receive

$$C_{Dj}^2(t) = \varrho_j^2(t)C_{Bj}^2(t) + C_{Aj}^2(t)(1 - \varrho_j(t)) + \varrho_j(t)[1 - \varrho_j(t)], \quad (14)$$

where $C_{Dj}^2(t)$, $C_{Bj}^2(t)$, $C_{Aj}^2(t)$ are time-dependent square coefficients of variation of interdeparture, service, and interarrival times, respectively. Packets leaving the node j according to the distribution $f_{Dj}(x,t)$ choose any node i with probability $r_{ji}(t)$ and the times between packets routed from node j to i has pdf $f_{ji}(x,t)$

$$f_{ji}(x,t) = f_{Dj}(x,t)r_{ji}(t) + f_{Dj}(x,t) * f_{Dj}(x,t)[1 - r_{ji}(t)]r_{ji}(t) +$$
$$f_{Dj}(x,t) * f_{Dj}(x,t) * f_{Dj}(x,t)[1 - r_{ji}(t)]^2 r_{ji} + \cdots \quad (15)$$

i.e. a packet leaving station j goes to station i with probability $r_{ji}(t)$ or with probability $1 - r_{ji}(t)$ it goes elswhere but the second goes to i with probability $r_{ji}(t)$, hence the gap has has pdf $f_{Dj}(x,t) * f_{Dj}(x,t)$ with probability $[1 - r_{ji}(t)]r_{ji}(t)$, etc, or, after Laplace transform

$$\bar{f}_{ji}(s,t) = \bar{f}_{Dj}(s,t)r_{ji}(t) + \bar{f}_{Dj}(s,t)^2[1 - r_{ji}(t)]r_{ji} +$$
$$+ \bar{f}_{Dj}(s,t)^3(1 - r_{ji}(t))^2 r_{ji} + \cdots$$
$$= \frac{r_{ji}(t)\bar{f}_i(s,t)}{1 - [1 - r_{ji}(t)]\bar{f}_i(s,t)} ,$$

and we compute the squared coefficient of variation

$$C_{ji}^2(t) = r_{ji}(t)[C_{Dj}^2(t) - 1] + 1,$$

and then the parameters of the input flow at station i are given by (12) and (16)

$$C_{Ai}^2(t) = \frac{1}{\lambda_{i-in}(t)} \sum_{j=1}^{M} r_{ji}(t)\lambda_{i-out}(t)[(C_{Di}^2(t)-1)r_{ji}(t)+1] + \frac{C_{0i}^2(t)\lambda_{0i}(t)}{\lambda_{i-in}(t)} , \quad (16)$$

where the parameters λ_{0i} and C_{0i}^2 refer to the flow coming to station i from outside of the network.

Eqs. (14), (16) form a system of linear equations yielding $C_{Ai}^2(t)$ and, in consequence, the diffusion parameters $\beta_i(t)$, $\alpha_i(t)$ for every node i. At each interval δ, functions $f_i(x, t; \psi_i)$ giving queue distributions at every station i for $t \in \delta$ are computed. Their valuest at the end of the interval yield, among others, the current utilisations ϱ_i used to determine the flow parameters and diffusion parameters for the next interval δ. This way the flow parameters change each δ and routing changes each $\Delta = n\delta$.

The pdf $f_{Ri}(x, t)$ of the current response time (waiting time plus service) is determined using the first passage time from the end of the queue to zero.

The first passage time of the diffusion process from x_0 to $x = 0$ has the density function [10]

$$\gamma_{x_0,0}(t) = \frac{x_0}{\sqrt{2\Pi\alpha t^3}} e^{-\frac{(\beta t+1)^2}{2\alpha t}},$$

with the Laplace transform

$$\bar{\gamma}_{x_0,0}(s) = e^{-x_0 \frac{\beta+\sqrt{\beta^2+2\alpha s}}{\alpha}}.$$

The starting point x_0 is determined by the function $f_i(x, t; \psi_i)$ hence

$$f_{Ri}(x, t) = \int_0^N \gamma_{\xi,0}(x) f_i(\xi, t; \psi_i) d\xi.$$

If $f_{Ri}(x, t)$ is the response time pdf at node i, then the response time pdf $f_R(x, t)$ for the path $1, \ldots, n$ of n stations is

$$f_R(x, t) = f_{R1}(x, t) * f_{R2}(x, t) * f_{R3}(x, t) * \cdots * f_{Rn}(x, t),$$

or

$$\bar{f}_R(x, s) = \prod_{i=1}^n \bar{f}_{Ri}(x, s).$$

The loss probability $p_{loss}(t)$ for same entire path may be computed from

$$1 - p_{loss}(t) = (1 - p_{N1}(t))(1 - p_{N2}(t))(1 - p_{N3}(t)) \ldots (1 - p_{Nn}(t)), \qquad (17)$$

where $p_{Ni}(t)$ is probabiliity that the queue at station i is saturated at time t, i.e. the diffusion process for this station is at time t at the barrier $x = N$.

4 Application to a SDN and a Numerical Experiment

Since the input and output hardware of a SDN forwarder is fast, the main component to be considered is the queue of packets waiting until the node identifies to which flow they belong and to 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.

Ler p 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 $(1 - p)$, the time to find the existing entry is uniformly distributed in $[T, KT]$ and having

$$\text{mean } (K + 1)T/2 \text{ and variance } (K^2 - 1)T^2/12.$$

The two cases define the first two moments $1/\mu$ and σ_B^2 of service time distribution in our $G/G/1/N$ diffusion model.
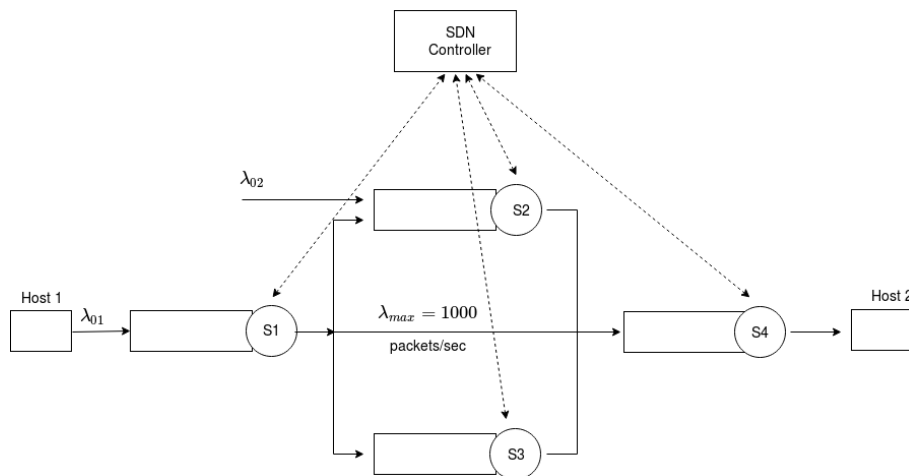
Fig. 1: The example network being considered

We consider a network composed of four switches, represented in Fig. 1. The network performance is investigated during 1 second. Host 1 is sending a flow of λ_{01} packets to Host 2. The intensity of the flow is changing in the range $500-2500$ packets/sec, see Fig. 2. If the flow is below 1000 packets per second, it is sent by the direct link $S1 - S4$, and if it exceeds the maximum capacity of this link, the surplus is sent in equal share by paths $S1 - S2 - S4$ and $S1 - S3 - S4$.

We assume at each station the buffers of $N = 100$ packets; in case of $S1$, $S2$, $S3$ the time to check one entry in the list of connections is $T = 8 \cdot 10^{-7}$ sec, and in $S4$ this time is twice shorter $T = 4 \cdot 10^{-7}$ sec. The number of entries $K = 950$, $p = 0$. It results in $\mu_1 = \mu_2 = \mu_3 = 2628.8$ packets/sec and $\mu_4 = 5257.6$ packets/sec. Squared coefficient of variation of service time C_{Bi}^2 is in all stations equal 0.33.

In the interval $[t = 0.450$ sec, $t = 0.705$ sec$]$, an additional flow λ_{02} of the intensity 1500 packets per second appears at station $S2$ and is also sent to Host 2 via $S2 - S4$. We consider three values of the squared coefficient of variation of interarrival times in the first flow: $C_{A1}^2 = C_{01}^2 = 1.02, 4.08$ and 8.16. The first value is obtained from our analysis of CAIDA data [1], and the others were chosen to see the network behaviour if the flow is more irregular. For the second flow, $C_{02}^2 = 1.02$.

The SDN controller alters the routing to balance the load of nodes every 100 msec, hence at $t = 500$ msec it reacts on the presence of the second flow and changes the routing r_{12} and r_{13}, see Fig. 3. In consequence, the load of stations $S2$ and $S3$ is changed, Fig. 4. After the end of the flow λ_{02} the initial routing is reestablished. The change of the utilisation influences the parameters of the output flows: as it is expressed by Eq. (14), higher the utilisation of a station i, closer its squared coefficient of variation of interdeparture times $C_{Di}^2(t)$ is to $C_{Bi}^2(t)$ and it is less dependent on $C_{Ai}^2(t)$. Fig. 5 displays the changes of $C_{Di}^2(t)$ following the pattern of input flows.

The transient solution of diffusion equations is computed in intervals of the length 10 msec, i.e. we have 100 intervals with fixed diffusion parameters; at the end of each the equations (12), (16) are solved to determine new parameters of flow for the single station models in the next interval. The diffusion density function obtained for any station i at the end of an interval gives initial conditions for the diffusion equation at the next one.

The model helps us to analyse the dynamics of every node. In Fig. 6 we see how the distribution of queue length (the queue is empty at the beginning, and it starts to be filled) at station $S1$ evolves with time. Even minimal values of the distributions are computed without numerical problems. As mentioned above, we used three different values of $C_{A1}^2(t)$, the squared coefficient of variation of interarrival times at station $S1$. In Fig. 7 the density function for $S1$ queue distribution is displayed for these values and makes evident their impact on the queue, note that the scale in Figs. 6, 7 is logarithmic.

The next figures display the impact of C_{A1}^2 on loss probability due to the full buffer at station $S1$, Fig. 8, and on the mean queue at this station, Fig. 9, following the changes of the flow intensity.

The next curves compare the loss probability, Fig. 10 (note here minimal values computed by the model), and mean queues for all four stations, Fig. 11, in case of $C_{A1}^2 = 1.04$. We may observe the changes in mean queues in $S2$ and $S3$ due to load balancing after the second flow becomes active. We see also, observing mean queues at $S1$ and $S2$, that transient periods may be longer than the time between the controller's decisions. The length of the transient time increases with a load of a station and variability of the input flow.

Figures 12 and 13 refer to station $S2$. We see here a weak impact of C_{A1}^2 on the mean queue. It is evident: as this station is mainly supplied by the second flow. However, if we consider loss probabilities which have here very small values and are displayed in logarithmic scale, the impact of C_{A1}^2 may be observed. It is better seen at station $S3$, Fig. 14, and in station $S4$, Fig. 15, because they

receive much more of the first flow. Note that for greater variabilities of the first flow, the path $S1 - S3 - S4$ becomes saturated, Fig. 16.

Let us also consider a **simple example of optimization**. Suppose as previously that station S1 is forwarding a flow λ_{01} packets to nodes S2 and S3. Station S2 is additionally receiving a flow of $\lambda_{02}^{(loc)}$ packets directly from the outside of the network. The controller is changing routing every $\Delta = 100$ msec and needs do determine routing probabilities for the nearest Δ, knowing current parameters of flows at the beginning of the interval, as well as the current queue distributions at S1, S2, S3, representing previous behaviour of the network. The goal is to minimise the mean backlog Ψ at S2 and S3 during Δ

$$\min_{r_{12},r_{13}} \left\{ \Psi = \frac{1}{\Delta} \int_0^\Delta \left[E[N_2(t)] + E[N_3(t)] \right] dt \right\}.$$

We compute $E[N_2(t)]$, $E[N_3(t)]$ for $t \in \Delta$ and minimize Ψ by choice of r_{12}, $r_{13} = 1 - r_{12}$, see Fig. 17.

5   Conclusions

The IoT provides large volumes of highly capillary traffic that includes data and video, which has stringent QoS and security constraints. These large volumes of traffic also create additional energy consumption in networks. Thus means are needed to distribute traffic dynamically so that security and QoS are assured, and energy consumpton is minimised.

Fortunately, the advent of SDN allows the implementation of smart adaptive routing [25] which allows network paths to change so that security incidents and traffic overloads can be accomodated by taking advantage of alternate paths. However this leads to an ineresting paradigm shift in network modeling which has been traditionally addressed via steady-state "long term" modelling techniques. However, when SDN intervenes dynamically to change paths and traffic levels, the network is seldom at steady-state so that optimisation must take transients into account.

To achieve this, this paper uses diffusion approximations for the performance evaluation of a network of SDN data plane switches with time-dependent routing. We show that this method is computationally operational, and that it can provide quantitative results for models with realistic parameter values.

Our analysis captures the interactions among the main parameters of the network, and numerical examples display the dependence of the queue lengths and queueing delays and their changing dynamics, on the flow intensity and variance of interarrival times.

Our approach confirms the fact that transient periods play a significant role in the performance of SDN networks, and in future work we will use it to analyse much larger networks.
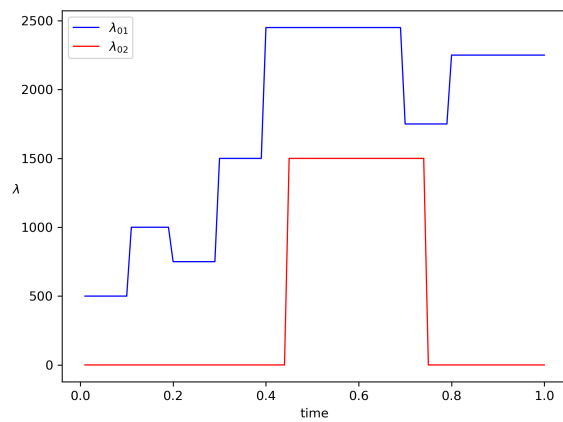
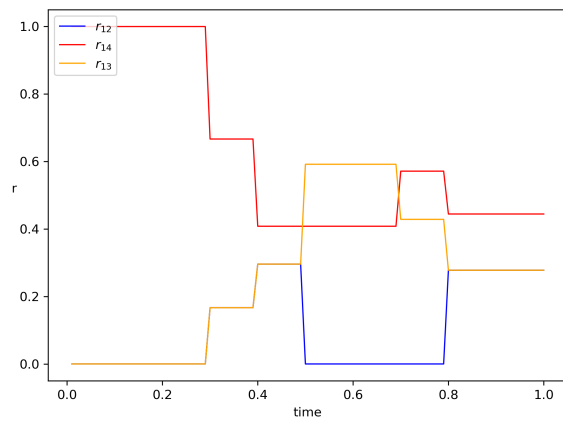Fig. 2: Input flows $\lambda_{01}(t)$, $\lambda_{02}(t)$, time in seconds

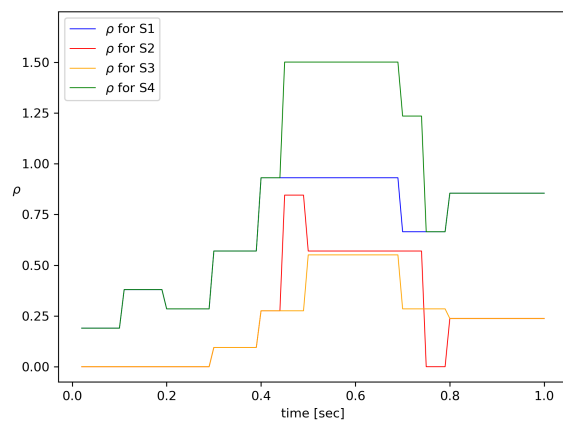Fig. 3: Routing probabilities $r_{12}(t)$, $r_{13}(t)$, $r_{14}(t)$
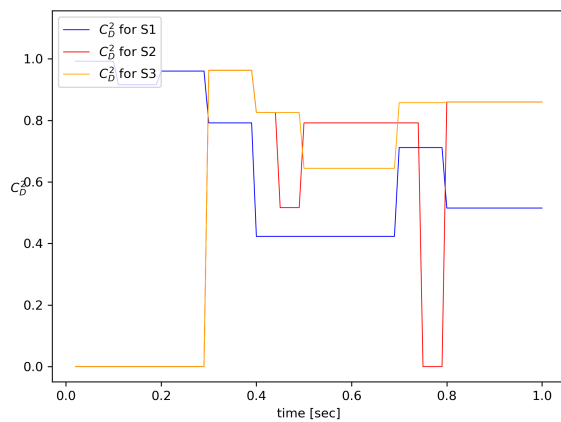

Fig. 4: $\lambda_i(t)/\mu_i$ for all stations

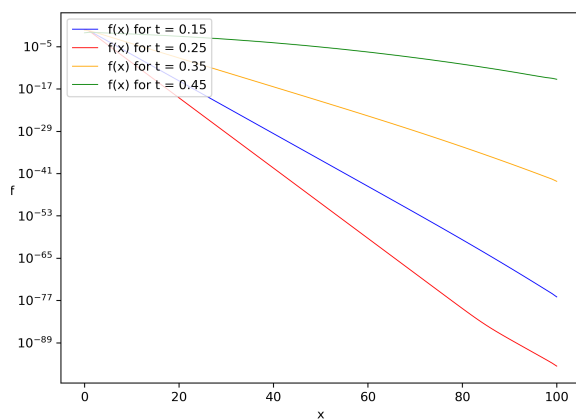Fig. 5: $S1$, $S2$, $S3$: squared coefficient of variation $C_{Di}^2(t)$ for the output flow, $C_{A1}^2 = 1.02$

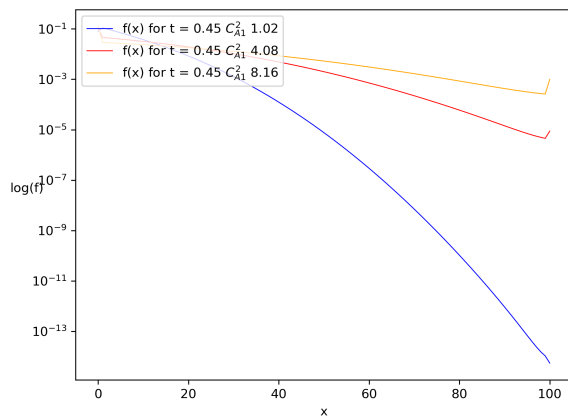Fig. 6: $S1$: density function $f_1(x, t; 0)$, $t = 0.15, 0.25, 0.35, 0.45$ sec, $C_{A1}^2 = 1.02$

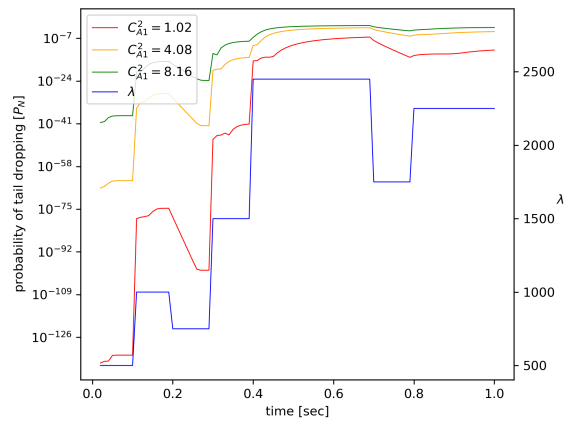Fig. 7: $S1$: density function $f_1(x, t; 0)$, $t = 0.45$ sec, for different C_{A1}^2

Fig. 8: $S1$: $p_N(t)$ for different C_{A1}^2
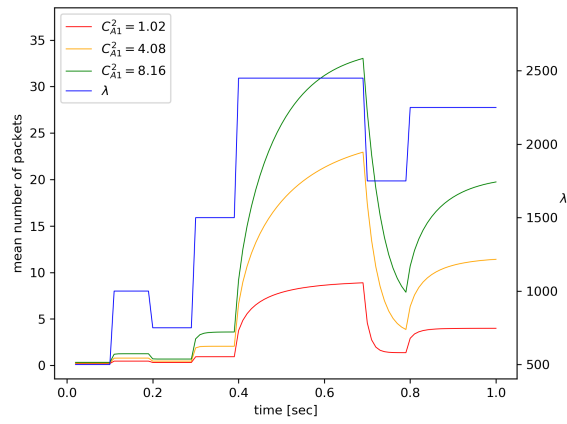
Fig. 9: $S1$: mean queue for different C_{A1}^2
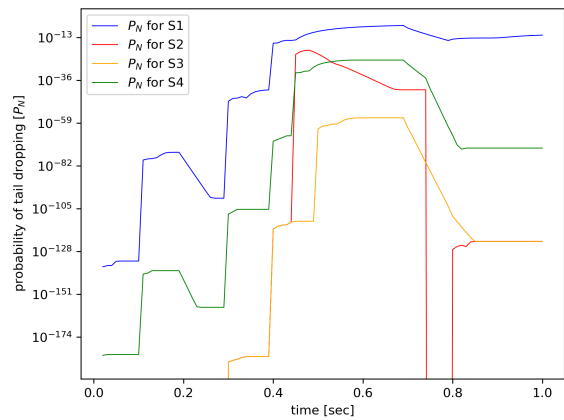
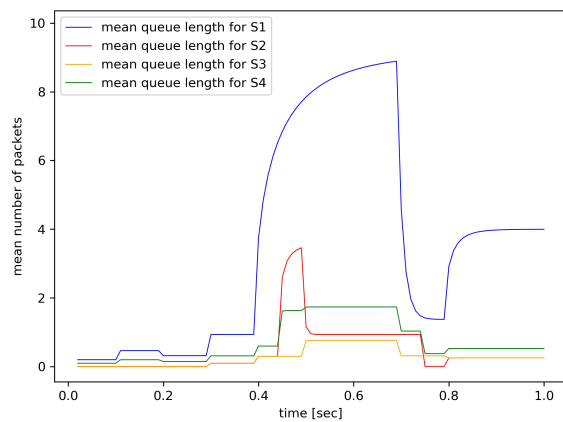Fig. 10: Stations $S1$, $S2$, $S3$, $S4$: $p_N(t)$, $C_{A1}^2 = 1.02$

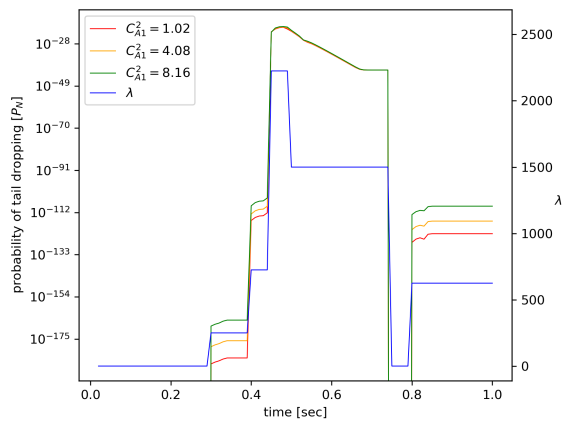Fig. 11: Stations $S1$, $S2$, $S3$, $S4$: mean queue, $C_{A1}^2 = 1.02$

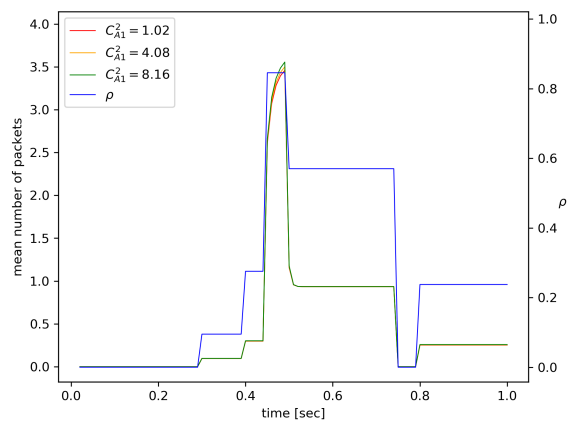Fig. 12: Station $S2$: $p_N(t)$ for different C_{A1}^2

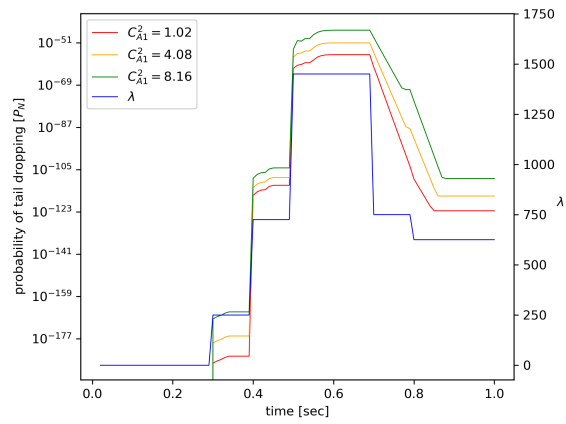Fig. 13: Station $S2$: mean queue for different C_{A1}^2

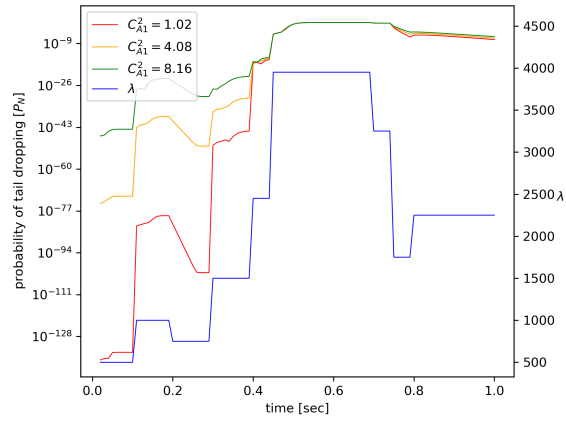Fig. 14: Station $S3$, $p_N(t)$ for different C_{A1}^2

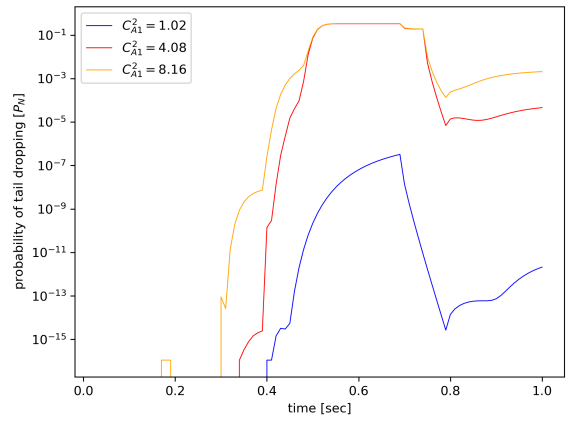Fig. 15: Station $S4$: $p_N(t)$ for different C_{A1}^2

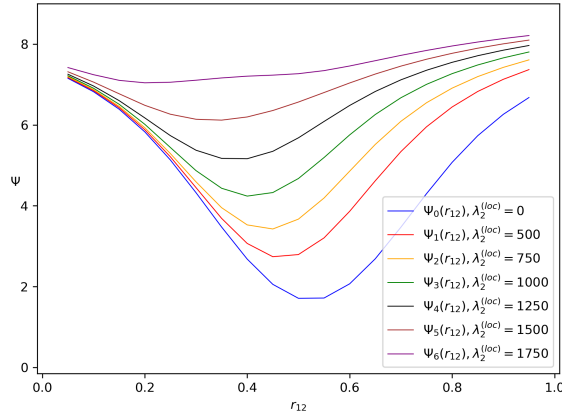Fig. 16: Total loss probability for path $S1 - S3 - S4$ for different C_{A1}^2

Fig. 17: Mean backlog Ψ during Δ as a function of routing probabilities r_{12}, $r_{13} = 1 - r_{12}$

Acknowledgements

References

1. https://data.caida.org/datasets/passive-2016/equinix-chicago/20160218-130000.UTC/
2. Ansell et al., J.: Making queueing theory more palatable to sdn/openflow-based network practitioners. In: Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium. pp. 1119–1124. IEEE, Istanbul, Turkey (2016). https://doi.org/10.1109/NOMS.2016.7502973
3. Augusto-Gonzalez et al., J.: From internet of threats to internet of things: A cyber security architecture for smart homes. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). pp. 1–6. IEEE (2019)
4. 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. pp. 68–74. IEEE, Berlin, Germany (2013). https://doi.org/10.1109/EWSDN.2013.18
5. Azodolmolky et al., S.: An analytical model for software defined networking: A network calculus-based approach. In: Proceedings of the IEEE Global Communications Conference. pp. 1397–1402. IEEE, Atlanta, USA (2013). https://doi.org/10.1109/GLOCOM.2013.6831269
6. Baldini et al., G.: Iot network risk assessment and mitigation: The seriot approach. In: Soldatos, J. (ed.) Security Risk Management for the Internet of Things: Technologies and Techniques for IoT Security, Privacy and Data Protection. p. 88–104. Now Publishers (2020). https://doi.org/DOI: 10.1561/9781680836837

7. Burke, P.J.: The output of a queuing system. Operations Research **4**(6), 699–704 (1956)
8. Buyya et al., R.: A manifesto for future generation cloud computing: Research directions for the next decade. ACM Computing Surveys (CSUR) **51**(5), 1–38 (2019)
9. Chesnais et al., A.: On the modeling of parallel access to shared data. Communications of the ACM **26**(3), 196–202 (1983)
10. Cox, R.P., Miller, H.D.: The Theory of Stochastic Processes. Chapman and Hall, London, UK (1965)
11. Czachórski, T.: A method to solve diffusion equation with instantaneous return processes acting as boundary conditions. Bulletin of Polish Academy of Sciences, Technical Sciences **41**(4) (1993)
12. Czachórski, T., Gelenbe, E., Kuaban, G.S., Marek, D.: Dynamics of software defined networks, diffusion model. Submitted to: Journal of Physics: Conference Series (2020)
13. Czachórski, T., Gelenbe, E., Kuaban, G.S., Marek, D.: Transient behaviour of a network router, accepted. In: Herencsár, N., Benedetto, F., Hosek, J. (eds.) Proceedings of International Conference on Telecommunications and Signal Processing TSP 2020, submitted. IEEE, Milano (2020)
14. Dobson et al., S.: A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **1**(2), 223–259 (2006)
15. Duda, A.: Diffusion approximations for time-dependent queueing systems. IEEE Journal on Selected Areas in Communications **4**, 905–918 (1986)
16. Fahmin, A., Lai, Y.C., Hossain, S., Lin, Y.D., Saha, D.: Performance modeling of sdn with nfv under or aside the controller. In: Proceedings of the 5th International Conference on Future Internet of Things and Cloud Workshops. pp. 211–216. IEEE, Prague (2017). https://doi.org/10.1109/FiCloudW.2017.76
17. Francois, F., Gelenbe, E.: Towards a cognitive routing engine for software defined networks. In: Proceedings of the 2016 IEEE International Conference on Communications. pp. 1–6. IEEE, Kuala Lumpur (2016). https://doi.org/10.1109/ICC.2016.7511138
18. Frötscher et al., A.: Improve cybersecurity of c-its road side infrastructure installations: the seriot-secure and safe iot approach. In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE). pp. 1–5. IEEE (2019)
19. Gelenbe, E.: On approximate computer system models. J. of the ACM **22**(2), 261–269 (1975)
20. Gelenbe, E.: Probabilistic models of computer systems part ii: Diffusion approximations, waiting times and batch arrivals. Acta Informatica **12**, 285–303 (1979), `https://doi.org/10.1007/BF00268317`
21. Gelenbe, E., Domanska, J., Czachórski, T., Drosou, A., Tzovaras, D.: Security for internet of things: The seriot project. In: 2018 International Symposium on Networks, Computers and Communications (ISNCC). pp. 1–5. IEEE (2018)
22. Gelenbe, E., Hebrail, G.: A probability model of uncertainty in data bases. In: 1986 IEEE Second International Conference on Data Engineering. pp. 328–333. IEEE (1986)
23. Gelenbe, E., Mang, X., Önvural, R.: Diffusion based statistical call admission control in atm. Performance evaluation **27**, 411–436 (1996)
24. Gelenbe, E., Pujolle, G.: The behaviour of a single-queue in a general queueing network. Acta Inf. **7**, 123–136 (1976). https://doi.org/10.1007/BF00265766, `https://doi.org/10.1007/BF00265766`

25. Gelenbe et al., E.: Self-aware networks that optimize security, qos and energy. Proceedings of the IEEE, accepted for publication **108**(7) (2020)
26. Hakiri et al., A.: Software-defined networking: Challenges and research opportunities for future internet. Computer Networks **75**, 453–471 (2014)
27. Kuźniar, M., Peresini, P., Kostić, D., Canini, M.: Methodology, measurement and analysis of flow tablee update characteristics in hardware openflow switches. Computer Networks **136**, 22–36 (2018)
28. Lai, Y.C., Ali, A., Hassan, M., Hossain, S., Lin, Y.D.: Performance modeling and analysis of tcp connections over software defined networks. In: Proceedings of the 2017 IEEE Global Communications Conference. pp. 1–6. IEEE, Singapore (2017). https://doi.org/10.1109/GLOCOM.2017.8254078
29. Mahmood, K., Chilwan, A., Osterbo, O., Jarschel, M.: Modelling of openflow-based software-defined networks: the multiple node case. IET Networks **4**(5), 278–284 (2015)
30. Marin, G.A., Mang, X., Gelenbe, E., Önvural, R.O.: Statistical call admission control. US Patent 6,222,824 (2001)
31. Miao, W., Min, G., Wu, Y., Wang, H., Hu, J.: Performance modelling and analysis of software defined networking under bursty multimedia traffic. ACM Transactions on Multimedia Computing, Communications, and Applications **12**(55), 24–36 (2018)
32. Natsiavas et al., P.: Comprehensive user requirements engineering methodology for secure and interoperable health data exchange. BMC medical informatics and decision making **18**(1), 85 (2018)
33. Nguyen-Ngoc, A., Lange, S., Geissler, S., Zinner, T., Tran-Gia, P.: Estimating the flow rule installation time od sdn switches when facing control plane delay. LNCS **10740**, 113=126 (2018)
34. Paul, S., Pan, J., Jain, R.: Architectures for the future networks and the next generatio internet: A survey. Computer Communications **34**, 2–42 (2011)
35. Rowshanrad et al., S.: A survey on sdn, the future of networking. Journal of Advanced Computer Science & Technology **3**(2), 232–248 (2014)
36. Singh et al., D.: Modelling software-defined networking: Software and hardware switches. Journal of Computer Network and Computer Applications **122**, 24–36 (2018)
37. Stehfest, H.: Numeric inversion of laplace transform. Communication of ACM **13**(1), 47–49 (1970)
38. Wang, L., Gelenbe, E.: Adaptive dispatching of tasks in the cloud. IEEE Transactions on Cloud Computing **6**(1), 33–45 (2018)
39. Wibowo, F.X., Gregory, M.A., Ahmen, K., Gomez, K.M.: Multi-domain software defined networking: Research status and challenges. Journal of Network and Computer Applications **87**, 32–45 (2017)
40. Wu, K.R., Liang, J.M., Lee, S.C., Tseng, Y.C.: Efficient and consistent flow update for software defined networks. IEEE Journal on Selected Areas in Communications **36**(3), 411–420 (2018)
41. Yoon, C., Park, T., Lee, S., Kang, H., Shin, S., Zhang, Z.: Enabling security functions with sdn: A feasibility study. Computer Networks **85**, 19–35 (2015)