

Transient Behaviour of a Network Router

Tadeusz Czachórski*, Erol Gelenbe, *Fellow IEEE**, Godlove Suila Kuaban*, and Dariusz Marek†

* Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
ul. Baltycka 5, 44-100 Gliwice, Poland

†Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science
ul. Akademicka 16, 44-100 Gliwice, Poland

Abstract—The broader use of Software Defined Network (SDN) controllers creates periodic changes in topology and traffic rates at routers that adapt the network to changes in network conditions. Thus the transient behaviour of network components, and in particular routers, is becoming of great interest. Since standard queueing models are difficult to analyze under time-varying conditions, we propose a tractable diffusion approximation for both the transient and steady-state behaviour of a network router. In particular, the analysis provides the steady-state and transient delay and packet loss probability as a function of traffic load and other characteristics. Using these results, we show that when SDN routers change the paths of flows frequently, the network’s behaviour may often be far from its steady-state behaviour. Therefore any network optimization conducted with the help of SDN should not be based on steady-state behaviour, but rather on some metric related to the network time dependent behaviour.

Index Terms—Network Routers, Internet Traffic, Quality of Service (QoS), Diffusion Approximation

I. INTRODUCTION

The large scale deployment of the IoT [1] together with Cloud Services [2], 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 traditional networks that must now ensure better security [3] of the traffic flows, acceptable quality of service, flexible network management, and energy optimization [4]. The introduction of artificial intelligence into network routing and management [5], [6] 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) [7] gives carriers, service providers and enterprises more significant control over the way traffic moves around networks [8], [9] 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 routers and network components becomes of great interest [10].

Recent studies have analyzed SDN networks to optimize steady-state performance using queueing [11], [12], [13], [14], [15], [16], [17] and network calculus [18], [19], [20]. Although

This research was supported by the SerIoT Research and Innovation Action, funded by the European Commission (EC) under the H2020-IOT-2017 Program, Grant Agreement 780139. The EC’s support does not constitute an endorsement of this paper, which reflects the views only of the authors.

these issues require the analysis of the transient behaviour of a router, 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 randomized 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 a diffusion approximation which offers 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. Let us note that diffusion predictions have been included in patented industrial telecommunication systems [21], [22]. 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 [23]. Thus, the approach we take in this paper allows us to predict the time it takes for a router to reach its new steady-state after the input traffic rates change, and to compute the packet loss probabilities in cases when they may be very small and hard to obtain by simulation.

II. DIFFUSION MODEL OF A ROUTER

We consider the simplified architecture of a router shown in Figure 1. One of the issues in the design of router architectures is the allocation of buffering resources [24].

When a packet arrives at the input port of a router, it is stored temporarily, provided it cannot be scheduled immediately for processing. An Arbiter then removes the packet from the input queue and stores it temporarily in an internal Packet Buffer while a copy of the header is sent to the Parser. The Parser extracts the header fields and creates a tuple which contains packet forwarding information and sends it to the Flow Match Unit which uses this information to look up the flow tables to match an entry [25]. If the match fails, the router fires off a packet-in message containing the full packet or its buffer ID to the connected SDN controller [26]. Otherwise, the packet is dropped. However, if a matching entry is found for the packet in the flow table, the flow rules installed by

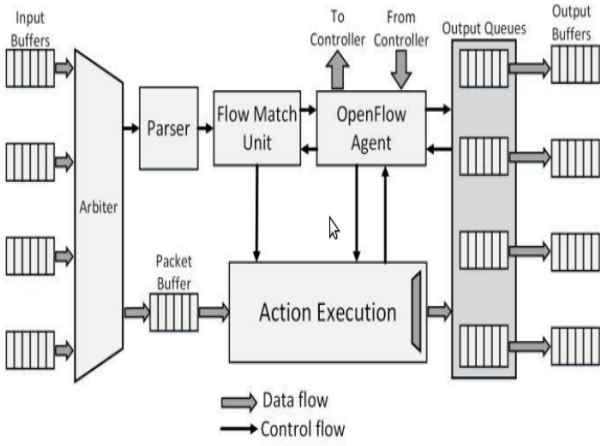


Fig. 1. An example of the architecture of a router [25].

the controller are applied, and the packet is removed from the internal Packet Buffer and forwarded through the back plane.

The input buffers and the Arbiter constitute the input queueing system, the internal Packet Buffers and the OpenFlow (OF) processing constitute the internal Packet Buffer queueing system and the output queues, buffers and transmitter constitute the output queueing system. We assume that the Arbiter is running at a faster speed (Gbps), such that there is little or no queueing at the input buffers and that for high-speed links, we have negligible queueing at the output queues. Therefore, the queueing model representation of the SDN forwarding node is composed of a single internal Packet Buffer queueing system.

Let p be the probability that the router's flow table does not contain the installed flow rule for a given arriving packet; it can happen because all the routers controlled by the same SDN controller are not updated in a synchronized manner. The absence of a flow forwarding rule for the arriving packet in the flow table will be discovered after the all K flow table entries have been checked, i.e. after time KT , where T is the time to check each flow table entry. Therefore with probability p the service time is constant, with zero variance. On the other hand, with probability $(1-p)$, if the flow matching rule for the packet exists in the router, it is in one of the K flow tables entries, and the time to find it is the time spent checking each flow table entry until the proper one is found. The distribution of this time is discrete uniform between T and KT , with mean $(K+1)T/2$ and variance $(K^2-1)T^2/12$, unless a more sophisticated search data structure is implemented (in which case the search time may be proportional to the logarithm of table size).

When a flow matching rule for a packet is found, the packet is sent to the appropriate output port for forwarding. The number of data packet frames transmitted per second depends on the transmission rate or speed and the distribution of the size of IP packets. For high transmission speeds, which is usual in core network routers, the delay in output ports may be negligible. This way the queueing model of the node is reduced to the model of the Packet Buffer. Since the buffer

is of finite capacity N , we will use a finite capacity diffusion approximation model type G/G/1/N where a discrete-state process $\{M(t), t \geq 0\}$ of the number of customers in the queue is replaced by an appropriate continuous diffusion process, $\{X(t), t \geq 0\}$. Its incremental changes $dX(t) = X(t+dt) - X(t)$ are normally distributed with mean βdt and variance αdt , where β , α are coefficients of the diffusion equation. For the process $M(t)$ the changes at ΔT have mean $(\lambda - \mu)\Delta T$ and variance $(\lambda^3\sigma_A^2 + \mu^3\sigma_B^2)\Delta T = (\lambda C_A^2 + \mu C_B^2)\Delta T$ where $1/\lambda$, $1/\mu$ are mean values of interarrival and service times distributions, σ_A^2 , σ_B^2 are their variances, and C_A^2 , C_B^2 their squared coefficients of variation. Therefore it is assumed $\beta = \lambda - \mu$ and $\alpha = \lambda C_A^2 + \mu C_B^2$.

Since the queue size must be positive, boundary conditions such as absorbing [27], reflecting [28] or elementary return barriers [29] are commonly used to constraint the diffusion process to the positive x -axis. Because of the finite capacity limited to N , the diffusion process should be limited to the $[0, N]$ interval. Therefore, the diffusion approximation model with elementary return barriers at $x = 0$ and $x = N$ is used [30]:

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

where $\delta(x)$ is Dirac delta function.

A. Steady State Delay

In steady state, when $\lim_{t \rightarrow \infty} p_0(t) = p_0$, $\lim_{t \rightarrow \infty} p_N(t) = p_N$, $\lim_{t \rightarrow \infty} f(x, t; x_0) = f(x)$, Eqs.(1) become ordinary differential ones and their solution, for $\rho = \lambda/\mu$, $\rho < 1$, can be expressed as [29]:

$$f(x) = \begin{cases} \frac{\lambda p_0}{-\beta} (1 - e^{zx}) & \text{for } 0 < x \leq 1, \\ \frac{\lambda p_0}{-\beta} (e^{-z} - 1) e^{zx} & \text{for } 1 \leq x \leq N-1, \\ \frac{\mu p_N}{-\beta} (e^{z(x-N)} - 1) & \text{for } N-1 \leq x < N, \end{cases} \quad (2)$$

where $z = \frac{2\beta}{\alpha}$, and the probability p_0 of empty buffer as well as the probability p_N of full buffer, i.e. packet tail dropping probability, are determined through normalization

$$\begin{aligned} p_0 &= \left\{ 1 + \rho e^{z(N-1)} + \frac{\rho}{1-\rho} [1 - e^{z(N-1)}] \right\}^{-1}, \\ p_N &= \rho p_0 e^{z(N-1)}. \end{aligned}$$

The values $p_0, f(1), f(2), \dots, f(N-1), p_N$ determine the distribution of the number of packets in the Packet Buffer. Figure 2 presents typical solutions $f(x)$ when we vary the parameter ρ and Figure 3 displays the packet loss probabilities $p(N)$ as a function of ρ . When the utilization is over 80%, the

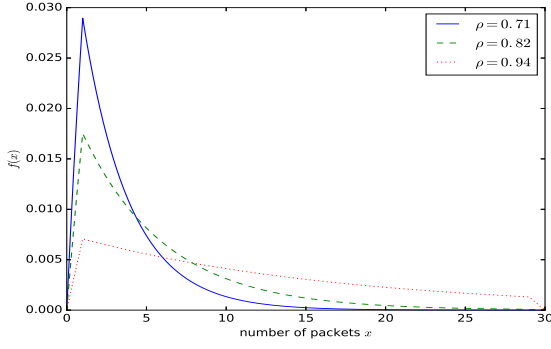


Fig. 2. Influence of the utilization ρ on the distribution of the queue size

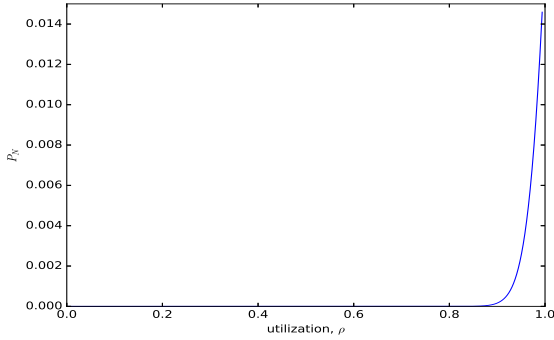


Fig. 3. Influence of the utilization on the probability of tail dropping of packets

probability of tail dropping of packets (buffer overflow) rises sharply.

In all numerical examples we assume $K = 950$, $T = 10^{-8}$ sec, $p = 0.01$. The mean and variance of service time distribution was determined as indicated above. For the interarrival time distribution, see Figure 4 we used data from CAIDA (Center for Applied Internet Data Analysis) traces [31], more precisely IPv4 packet interarrival times from the Equinix Chicago link collected during one hour on 18 February 2016, having more than 22 millions of packets belonging to over 1.17 millions of IPv4 flows. The obtained coefficient of variation of the interarrival times was 1.02. The mean value is scaled to assure various values of λ .

To determine the delay distribution, we will use the notion of the first passage time which corresponds to the queueing time.

The probability density function $\phi(x, t; x_0)$ for a diffusion process that starts at $t = 0$ from $x = x_0$ (initial queue size seen by an arriving packet when it joins the queue) and ends when it attains the absorbing barrier at $x = 0$ is [27]

$$\phi(x, t; x_0) = \frac{e^{\frac{\beta}{\alpha}(x-x_0) - \frac{\beta^2}{2\alpha}t}}{\sqrt{2\pi\alpha t}} \left[e^{-\frac{(x-x_0)^2}{2\alpha t}} - e^{-\frac{(x+x_0)^2}{2\alpha t}} \right]. \quad (3)$$

The density function of first passage time from $x = x_0$ to

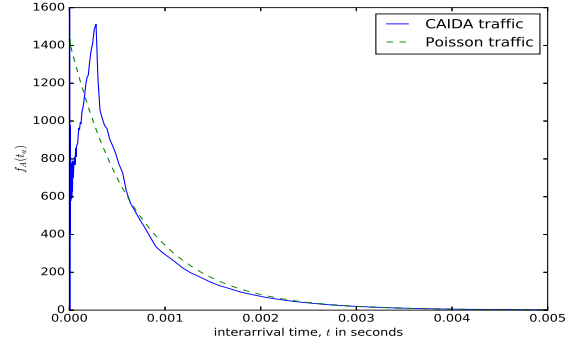


Fig. 4. The interarrival times of IP traffic.

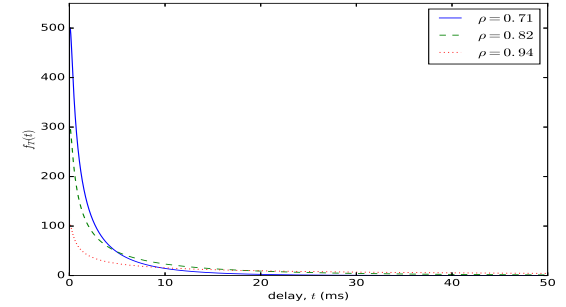


Fig. 5. Influence of the utilization on the distribution $f_T(x)$ of the delay

$x = 0$ is

$$\begin{aligned} \gamma_{x_0,0}(t) &= \lim_{x \rightarrow 0} \left[\frac{\alpha}{2} \frac{\partial}{\partial x} \phi(x, t; x_0) - \beta \phi(x, t; x_0) \right] \\ &= \frac{x_0}{\sqrt{2\pi\alpha t^3}} e^{-\frac{(x_0-\beta)^2}{2\alpha t}}. \end{aligned} \quad (4)$$

and after normalization of $\gamma_{x_0,0}(t)$ for $\beta < 0$ we have

$$\int_0^\infty \gamma_{x_0,0}(t) dt = e^{\frac{2x_0\beta}{\alpha}}, \quad (5)$$

and the probability density function (PDF) of the first passage time of the diffusion process from $x = x_0$ to $x = 0$, for $\beta < 0$ is

$$\gamma_{x_0,0}(t) = \frac{x_0}{\sqrt{2\pi\alpha t^3}} e^{-\left[\frac{2x_0\beta}{\alpha} + \frac{(x_0-\beta)^2}{2\alpha t} \right]}. \quad (6)$$

Suppose that a newly arrived packet sees x packets with density $f(x)$ given by Eq. (2), then the density of the delay is

$$f_T(t) = \int_0^\infty \left[\frac{x}{\sqrt{2\pi\alpha t^3}} e^{-\left[\frac{2x\beta}{\alpha} + \frac{(x-\beta)^2}{2\alpha t} \right]} \right] f(x) dx. \quad (7)$$

Figure 5 presents the PDF $f_T(t)$ of the delay for a few values of utilization ρ .

B. Transient delay modeling

In the case of steady-state analysis, the first two moments of the interarrival and service times used to calculate the diffusion parameters are constant. However, due to the unpredictable

characteristics of user traffic and the use of adaptive routing protocols such as the self-aware routing protocol used in SerIoT SDN core network, the characteristics of the traffic arriving at the input and output buffers are dynamic. It requires the transient delay analysis within short time intervals, where the diffusion parameters are constant only with these interval time interval.

Consider a diffusion process with two absorbing barriers at $x = 0$ and $x = N$, that started at $t = 0$ from $x = x_0$ and that its probability density function $\phi(x, t; x_0)$ has the following form [27]

$$\phi(x, t; x_0) = \begin{cases} \delta(x - x_0) & \text{for } t = 0, \\ \frac{1}{\sqrt{2\pi\alpha t}} \sum_{n=-\infty}^{\infty} \{a(t) + b(t)\} & \text{for } t > 0, \end{cases} \quad (8)$$

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

Suppose that the diffusion process starts at point ξ with PDF $\psi(\xi)$, $\xi \in (0, N)$, $\lim_{\xi \rightarrow 0} \psi(\xi) = \lim_{\xi \rightarrow N} \psi(\xi) = 0$, then the PDF of the process has the form

$$\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi) \psi(\xi) d\xi. \quad (9)$$

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

$$\bar{\phi}(x, s; x_0) = \frac{\exp\left[\frac{\beta(x-x_0)}{\alpha}\right]}{A(s)}. \quad (10)$$

$$\cdot \sum_{n=-\infty}^{\infty} \left\{ \exp\left[-\frac{|x - x_0 - x'_n|}{\alpha}\right] A(s) \right. \\ \left. - \exp\left[-\frac{|x - x_0 - x''_n|}{\alpha}\right] A(s) \right\}, \quad (11)$$

where $A(s) = \sqrt{\beta^2 + 2\alpha s}$.

Since the transient solution of equation (1) is not analytically tractable, the probability density function $f(x, t; \psi)$ of the diffusion approximation process with elementary returns boundaries can be obtained numerically. It is composed of the function $\phi(x, t; \psi)$, which is the probability density function of the diffusion process with absorbing barriers at $x = 0$ and $x = N$ and the functions $\phi(x, t - \tau; 1)$ and $\phi(x, t - \tau; N - 1)$ which are probability density functions of the diffusion processes that started at time $\tau < t$ at points $x = 1$ and $x = N - 1$ with densities $g_1(\tau)$ and $g_{N-1}(\tau)$ with instantaneous jumps [32][33][34]

$$f(x, t; \psi) = \phi(x, t; \psi) + \int_0^t g_1(\tau) \phi(x, t - \tau; 1) d\tau \quad (12) \\ + \int_0^t g_{N-1}(\tau) \phi(x, t - \tau; N - 1) d\tau.$$

The densities $g_1(t)$ and $g_N(t)$ may be expressed with the use of functions $\gamma_0(t)$ and $\gamma_N(t)$:

$$g_1(\tau) = \int_0^\tau \gamma_0(t) l_0(\tau - t) dt \\ g_{N-1}(\tau) = \int_0^\tau \gamma_N(t) l_N(\tau - t) dt, \quad (13)$$

where $l_0(x)$, $l_N(x)$ are the densities of sojourn times at $x = 0$ and $x = N$ respectively, while $\gamma_0(t)$ and $\gamma_N(t)$ are the probability densities that at time t the process enters to $x = 0$ or $x = N$ are

$$\gamma_0(t) = p_0(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,0}(t) \\ + \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau) d\tau \\ + \int_0^t g_{N-1}(\tau)\gamma_{N-1,0}(t - \tau) d\tau, \\ \gamma_N(t) = p_N(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,N}(t) \\ + \int_0^t g_1(\tau)\gamma_{1,N}(t - \tau) d\tau \\ + \int_0^t g_{N-1}(\tau)\gamma_{N-1,N}(t - \tau) d\tau, \quad (14)$$

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.

$$\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \left[\frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x} - \beta \phi(x, t; 1) \right], \quad (15)$$

and for absorbing barriers,

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

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

The Laplace transform of the density function $f(x, t; \psi)$ is

$$\bar{f}(x, s; \psi) = \bar{\phi}(x, s; \psi) + \bar{g}_1(s) \bar{\phi}(x, s; 1) \\ + \bar{g}_{N-1}(s) \bar{\phi}(x, s; N - 1), \quad (16)$$

and the densities $\bar{g}_1(s)$, $\bar{g}_{N-1}(s)$ are obtained from (13), (14), (15) after their Laplace transform. The probabilities that at time t the process has the value $x = 0$ or $x = N$ are

$$\bar{p}_0(s) = \frac{1}{s} [\bar{\gamma}_0(s) - \bar{g}_1(s)], \quad (17) \\ \bar{p}_N(s) = \frac{1}{s} [\bar{\gamma}_N(s) - \bar{g}_{N-1}(s)].$$

The above solution gives the transient distribution of the queue length and the transient probability of packet losses when the buffer is full. The original functions of the Laplace transforms can be obtained numerically using Stehfest's algorithm [35], valid for constant diffusion parameters, i.e. constant traffic intensity λ . Therefore it is used for time intervals within which parameters are constant and the solution at the end of such interval serves as the initial condition, i.e. function $\psi(\xi)$

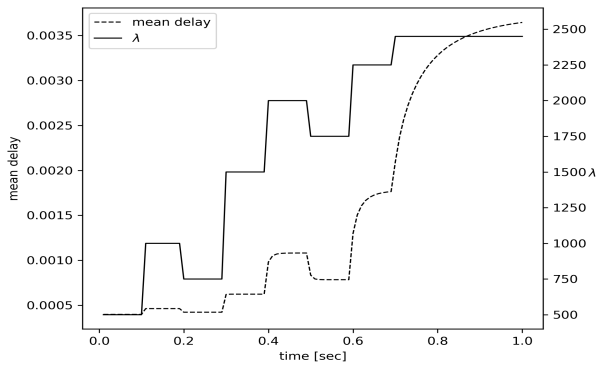


Fig. 6. The effect of abrupt changes the traffic arrival rate λ on the time dependent behaviour of the expected packet delay at the router.

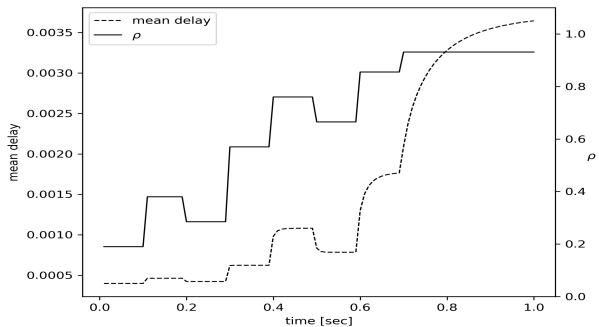


Fig. 7. Impact of abrupt changes in the queue utilisation ρ on the time dependent behaviour of the expected packet delay.

in (9) in the next interval with different parameters. The mean queueing delay was determined with the use of Little's formula but the first passage time approach is also possible.

III. NUMERICAL EXAMPLES

Based on the previous analysis, we have examined the effect of changes in the levels of arriving traffic rates to a router which may result from path changes created by SDN controllers. We have assumed that the router's packet buffer is partitioned into $N = 100$ packet sections where each section is reserved for a given active packet flow. When a packet arrives to the buffer, the time it takes to scan the table that contains the list of flows is assumed to be uniformly distributed with average value $S = 0.038ms$ and squared coefficient of variation is 0.33; though these values will vary with the router hardware, they are compatible with those of existing equipment.

In Figure 6 the arriving traffic rates of a given flow vary in the range of 500 to 2500 packets per second, and the traffic level λ changes approximately every 100ms reflecting relatively frequent path changes. We notice that, while at low traffic values the mean delay of a packet closely matches the steady-state value which is reached rapidly, at high values the mean delay always remains in its transient state so that the steady-state value is a poor predictor of the actual delay experienced by packets. Similar results, for another sequence

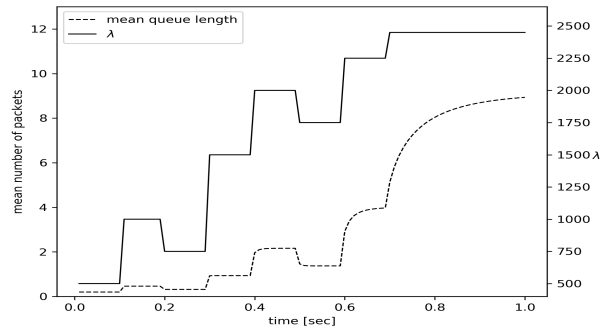


Fig. 8. The effect of abrupt changes in the packet arrival rate on the time-dependent behaviour of the expected queue length.

of changes in traffic arrival rate, are shown in Figure 7, where the time-dependent mean packet delay is plotted against the queue utilisation $\rho = \lambda S$. Confirming the results of the previous figure, we see here too that as ρ increases, the mean packet delay through the router never actually attains its steady-state value.

IV. CONCLUSIONS

Networks that are controlled by SDN are subject to frequent changes in network state as the SDN controller modifies paths in the network so as to optimize Quality of Service, Security or Energy Consumption. These frequent changes may imply that rather than running at a steady-state regime, the network will mostly find itself in transitory states. Diffusions approximations are far more convenient for the transient analysis of service systems, rather than queueing networks and discrete event simulation. Therefore we examine the transient behaviour of a network router with a diffusion approximation model to evaluate both the transient and steady-state performance of a network router, in order to predict packet delay through the router, and its packet loss probability.

The diffusion approximation shows that the transitory behaviour of each router depends on the load which results from the arrival rate of packets and the service process for each packet leaving the router. The service process in turn depends on the number of flows that the router handles because a possibly large flow table has to be searched to determine each incoming packet's outgoing link [17]. Thus our model also takes into account the dependence of the service time for each outgoing packet on the size of the flow table.

Our analysis allows us to predict the time dependent behaviour of important performance metrics such as the mean delay experienced by a packet at the router, the packet queue length for each flow, and the packet loss probability. It also shows that the time dependent behaviour tends much more slowly to its steady-state when the system is more heavily loaded. Numerical examples based on the analysis are also presented to illustrate these insights.

As a consequence of our analysis we see that future work should consider SDN based network optimization techniques

that focus both on the transient and steady-state behaviour, because the steady-state may not be attained in many cases.

Future work should also compare these theoretical results with measurements and investigate the performance implications of the detailed interaction of SDN controllers with their connected routers. In addition, we hope to use diffusion approximations to evaluate the performance of networks or systems where the objective of the controls is to optimize the performance of the system [36].

REFERENCES

- [1] S. K. Lee, M. Bae, and H. Kim, "Future of iot networks: A survey," *Applied Sciences*, vol. 7, no. 10, 2017.
- [2] 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.
- [3] 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.
- [4] 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>
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] T. Czachórski and K. Grochla, "Diffusion approximation models for cloud computations with task migrations," in *IEEE International Conference on Fog Computing (ICFC'19)*. IEEE Computer Society, 2019, pp. 27–30.
- [11] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "Modelling of openflow-based software-defined networks: the multiple node case," *IET Networks*, vol. 4, no. 5, pp. 278–284, 2015.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [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] 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.
- [21] G. A. Marin, X. Mang, E. Gelenbe, and R. O. Onvural, "Statistical call admission control," *IEEE communication Letters*, vol. 222, no. 824, 2001.
- [22] E. Gelenbe, X. Mang, and R. O. Onvural, "Diffusion based statistical call admission control in atm," *Performance Evaluation*, vol. 27, no. 12, pp. 411–436, 1996.
- [23] N. Bisnik and A. A. Abouzeid, "Queueing network models for delay analysis of multihop wireless ad hoc networks," *Ad Hoc Networks* 7, vol. 7, no. 1, pp. 79–97, 2009.
- [24] D. Padua, Ed., *Router Architecture*. Boston, MA: Springer, US, 2011, pp. 1748–1748.
- [25] S. Wijeratne, A. Ekanayake, S. Jayaweera, D. Ravishan, and A. Pasqual, "Scalable high performance router architecture on fpga for core networks," in *Proceeding of the 2019 ACM/SIGDA International Symposium*. ACM, 2019.
- [26] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of openflow-based software-defined networks based on queueing model," *Computer Networks*, vol. 102, pp. 172–185, 2016.
- [27] R. P. Cox and H. D. Miller, *The Theory of Stochastic Processes*. London, UK: Chapman and Hall, 1965.
- [28] S. Chiamsiri and S. Moore, "Accuracy comparisons between two diffusion approximations for mx/g/1 queues - instantaneous return vs. reflecting boundary," Atlanta, 1977.
- [29] E. Gelenbe, "On approximate computer systems models," *Journal of the ACM*, vol. 22, no. 2, pp. 261–269, 1975.
- [30] —, "Probabilistic models of computer systems part ii: Diffusion approximations, waiting times and batch arrivals," *Acta Informatica*, pp. 285–303, 1979.
- [31] [Online]. Available: <https://data.caida.org/datasets/passive-2016/equinix-chicago/20160218-130000.UTC/>
- [32] T. Czachórski and F. Pekergin, "Diffusion approximation as a modelling tool," in *Network Performance Engineering*, D. D. Kouvatso, Ed., vol. 5233. Springer, 2011, pp. 447–476.
- [33] T. Czachórski, "Queueing models for performance evaluation of computer networks-transient state analysis," in *Analytic Methods in Interdisciplinary Applications*, M. R. V. Mityushev, Ed., vol. 116. Springer, 2014, pp. 51–80.
- [34] T. Czachórski, M. Nycz, and T. Nycz, "Modelling transient states in queueing models of computer networks: A few practical issues," in *Distributed Computer and Communication Networks*, V. V. et al., Ed., vol. 279. Moscow, Russia: Springer, 2013, pp. 58–72.
- [35] H. Stehfest, "Numeric inversion of laplace transform," *Communication of ACM*, vol. 13, no. 1, pp. 47–49, 1970.
- [36] L. Wang and E. Gelenbe, "Adaptive dispatching of tasks in the cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 33–45, 2018.