Looking Beyond the First Moment: Analysis of Packet-related Distributions in P4 Systems with Controller Feedback

Nicolai Kröger*, Hasanin Harkous†, Fidan Mehmeti*, Wolfgang Kellerer*
**Technical University of Munich*, firstname.lastname@tum.de
†*Nokia*, firstname.lastname@nokia.com

***Abstract*—The ubiquitous spread of programmable data planes has been conditioned by the development and use of domain specific languages. One such very convenient programming language is P4, which enables devices, like switches, to be configurable and protocol-independent, making it a perfect match for Software Defined Networks. Therefore, analyzing the metrics of interest in such a network is of paramount importance to understand what actually happens in the system. However, while previously there were studies dealing with performance analysis on P4-enabled systems, these were mostly bounded to obtaining the first moment of the metrics of interest. This does not provide a full picture of how P4-programmable switches operate. Hence, in this paper, we provide an analysis of the distributions of the metrics of interest in the system, modeling its behavior as a queueing network. We provide arguments as to why a normal distribution can mimic the service time distribution of the data plane. We consider the behavior under different distributions of the service times in the control plane. Results show that the variance of the sojourn time tends to decrease when a higher number of packets is sent back to the controller, which is more emphasized with the medium-rate and slow controllers, where the coefficient of variation can be reduced by at least 35%.**

***Index Terms*—P4, SDN, Queueing networks with feedback, Performance Evaluation.**

I. INTRODUCTION

Systems with feedback controllers are very often encountered in different networks. Among the most well known is the Software-Defined Network (SDN) architecture [1], in which the data plane is separated from the control plane, with the goal to improve the overall network performance and monitoring.

Programmable data planes arose as one of the most prominent features of SDNs, enabling multifaceted ways to forward data packets while supporting a wide range of protocols. The P4 programming language [2] is one of the most well known enablers of these operations, and is used to configure the forwarding actions in switches.

Since P4 is experiencing a wider spread every day, it is very important to understand the achievable performance and limitations it provides in order to assess its suitability for certain use cases. While the performance of P4-assisted systems has been investigated previously, either through measurements [3] or theoretical analysis [4], the obtained results are bounded to be expressed only in terms of the first moments, i.e., averages. Lacking the knowledge of complete statistics of the metrics of interest prevents us from having a broader view of the (internal) system behavior, and consequently, understanding all the intricacies of operation in different regimes. Moreover, we are not able to understand worst-case scenarios in terms of the latency of such systems.

However, inferring all those details is cumbersome. The first reason is that the known theoretical models work well only under some special cases [5], but most of the time these models do not capture reliably the actual system behavior; in the other cases, only non-closed form approximations can be obtained, which do not provide a clear picture on the parameter dependency of the metrics of interest. On the other hand, performing simulations or measurements for all the possible scenarios is impossible. Therefore, a combination of both worlds would be the most viable approach.

There are some important research questions that arise related to the operation of a P4-based system:

- What is the distribution of the time a packet spends in the data plane? What about the entire sojourn distribution?
- What is the impact of the controller on the sojourn time of the packet?

To answer the aforementioned questions, in this paper we model the behavior of the system with a queueing network with partial feedback, in the sense that a packet can be sent to the controller at most once. We use simulations to analyze the behavior of the system when varying a wide range of system parameters. The evaluation focuses on the distribution of the inter-arrival and departure processes in the data plane and control plane. The observations we obtain in this paper are important as they can be used to infer interesting conclusions in terms of the system behavior and worst-case scenarios, including entire distributions of the metrics of interest, and not only information about the first moment. The main messages of this paper are two-fold. The first is that the more packets are forwarded to the controller, the lower the variability of the packet sojourn times in the system is. The second is that increasing the external arrival rate to the system increases the variability of the sojourn times for the fast deterministic

This work was supported in part by the Federal Ministry of Education and Research of Germany (BMBF) under the project "6G-Life", with project identification number 16KISK002.

controller, whereas the opposite effect is observed for the medium-rate hyper-exponential controller.

Specifically, our main contributions are:

- We analyze the behavior of the distributions of the sojourn time of the packets in a P4-based SDN system, using a special type of queueing networks where a packet can at most once go back to the controller.
- We provide insights on the sojourn time for different controller behaviors, capturing jointly the speed at which they operate and the distribution of the service time.
- Using a realistic simulator, with input data from measurements on the data plane, we obtain some other engineering insights about the system behavior under different scenarios.

The remainder of this paper is organized as follows. In Section II, we present some background information. This is followed by the problem formulation in Section III. The analysis for the fast controller is presented in Section IV, followed by the analysis for the medium-rate controller in Section V. The case of slow controller is presented in Section VI. A comparison between the three controllers is shown in Section VII. Some related work is discussed in Section VIII. Finally, Section IX concludes the paper.

II. P4 Background

With the emergence of programmable data planes, P4 [2] as a domain specific language, suitable for forwarding planes, has been introduced. It enables programmable devices to be configurable, protocol-independent and also target-independent. In order to achieve that, the language abstracts an underlying hardware architecture. Vendors only need to provide a compiler for their specific hardware. Thus, we can say that P4 is almost oblivious to the underlying hardware.

While the actual realization might change slightly for different hardware implementations, the abstracted P4 processing pipeline consists of three main stages: the *parser*, *match-action tables*, and *de-parser*. When arriving to a P4 switch, the defined headers of a packet are parsed, i.e., extracted. Additionally, metadata such as the ingress port is extracted. The parsed headers and the metadata are then used as key values and compared to the entries in match-action tables. If an entry for the key exists, the corresponding action is executed. This can include changing metadata such as the egress port, modifying or removing headers or any other programmed action. If there is no match, a pre-defined action can be executed. This includes actions like dropping the packet or sending it to the controller for further processing. If the packet is sent to the control plane, the controller adds a table entry to the match-action tables in the data plane to provide processing information for that packet. Then, the packet is re-injected to the data plane. After the packet is processed according to the matched action, packet headers are assembled together with the payload and the packet is sent to the egress port.

A P4 program defines the packet processing in such a pipeline. Hardware-specific compilers then adapt the description to the hardware architecture of P4 targets. P4 targets can be software-based such as BMv2 [6] and T4P4S [7], or hardware-based such as Netronome SmartNICs [8] and FPGA implementations [9].

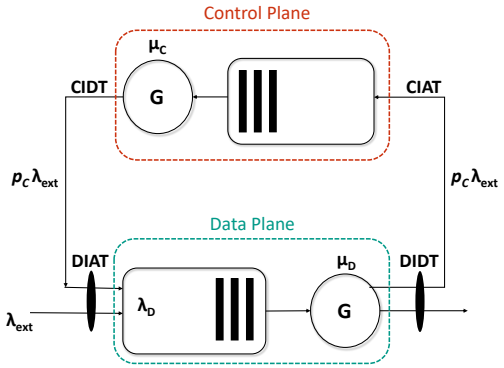


Fig. 1: The model for a general forwarding queue with controller feedback.

III. Problem Formulation

In this section, we first provide a general overview of the P4-device model. Then, we describe in detail the data plane, which is followed by a discussion of the control plane model. Finally, we define the processes of our interest in the system.

A. Performance Model of P4 devices

Using the SDN paradigm, the P4 concept decouples the data plane and control plane. In the former, packets are being processed according to the description of a P4 program. The control plane defines *which* packet is *how* executed by inducing rules to the data plane, i.e., to the match-action tables. One SDN controller can control several data planes.

In order to properly design a network with data planes and controllers, it is important to understand the behavior and limitations of P4 devices. One important performance metric is the *sojourn time* of the packets. As a first step, the interaction of a single data plane with one controller is assumed. Fig. 1 shows the model for such a system, where the data plane on the lower part and the control plane on the upper part are modeled as queues, each with a waiting buffer (of infinite size) and a server. The waiting queue abstracts all buffers in a device which stores packets until they can be processed. Both queues together form a network, known as *queueing networks with feedback* [5].

The life cycle of a packet in this model begins with the arrival at the switch, which is assumed to follow a Poisson process with rate λ_{ext}. After it has been processed according to the respective program with a service rate of μ_D, the packet leaves the switch. The packet either leaves the system completely or it is sent back to the controller with probability p_C. The latter case usually occurs when it is not known how to handle a specific packet in the switch and therefore needs the controller involvement. If the control plane interaction is needed for the packet, it is being processed by the controller with a service rate of μ_C and sent back afterwards to the data plane. As the controller updates the rules for packet handling

in the data plane, the packet after arriving again at the data plane is being processed successfully according to the new installed rule and leaves the system afterwards. So, each packet can "visit" the controller at most once.

Analyzing queueing networks and obtaining closed-form solutions is possible only for Jackson networks [10], i.e., when all the processes in the system are characterized by the memoryless property. On the other hand, in this work we go one step further and consider a more realistic setup, by assuming non-exponential distributed service times for the data plane and generally distributed service times on the control plane. Both components are described in more details in the following subsections. On top of that, we are also interested in the distributions of the metrics of interest, and not only in their first moments.

These assumptions and requirements make the derivation of analytical solutions in closed-form infeasible. Hence, in this work we focus on gaining insights on the behavior to the best possible extent analytically, and when that is not possible, using simulations. Our main metric of interest is the sojourn time, i.e., the overall time a packet spends in the system. In our model, a packet either leaves the system directly after being processed or is forwarded to the controller with a probability p_C. Thus, the sojourn time for the former packets is the time spent at the data plane. As the latter packets are sent to the data plane again after being processed at the controller, their sojourn time consists of the time spent on the control plane and the (two) times spent at the data plane.

B. Data Plane

The data plane follows the First In, First Out (FIFO) service principle and has an infinite queue size. The arrival process to the data plane consists of the combination of the external Poisson process with rate λ_{ext} and the feedback process from the controller with rate $p_C \cdot \lambda_{ext}$. Thus, the overall arrival rate λ_D to the data plane can be expressed as

$$\lambda_D = \lambda_{ext} + \lambda_{ext} \cdot p_C. \tag{1}$$

The arrival process to the data plane is not a Poisson process despite the fact that external inter-arrival times are exponentially distributed. The reason lies in the fact that the feedback is not independent from external arrivals, leading to non-independent increments. Hence, arrivals at the data plane do not follow a Poisson process.

We assume service times in the data plane are subject to a normal (Gaussian) distribution, with mean S_D (the service rate $\mu_D = \frac{1}{S_D}$), and standard deviation σ_D. The rationale behind making this assumption stems from the fact that each P4 program consists of certain atomic constructs, such as parsing header through which each packet must traverse. Moreover, in [3] it is shown that the number of these steps the packet undergoes can be quite large. Hence, even if the time the packet spends in each stage is not i.i.d. (but independent), due to the Lindeberg's condition [11], when the impact of a single stage in terms of the variance can be neglected compared to the overall process, the service time can be approximated with a normal distribution, in line with Central Limit theorem [12].

C. Control Plane

The control plane also follows the FIFO service principle and the queue size is infinite. The arrival process to the control plane is a part of the data plane departure process. Its rate λ_C is a fraction of the external arrival stream, i.e., given by

$$\lambda_C = p_C \cdot \lambda_{ext}. \tag{2}$$

It is worth stipulating again that a packet can at most once visit the controller. We refer to the packets that are being forwarded to the controller and then sent back to the data plane as *feedback packets*.

The service times at the controller are generally distributed with mean S_C, standard deviation σ_C, and service rate $\mu_C = \frac{1}{S_C}$. As there are no publicly-available data from measurement campaigns of service times at the controllers but rather only average times are provided [13], to consider all the possible scenarios, we split the operation mode of the controller in three regions where we model the service time with different distributions and service rates. In the first (Section IV), we assume a deterministic service time, which could correspond to fast controller where all the packets would spend (almost) the same time. In the second scenario of interest, we consider a moderate-speed controller (across all packets), in which some packets spend less time, while some others (much fewer of them) spend more time, resulting in a medium rate. Based on the provided description, we assume that the service time in this controller undergoes a hyper-exponential distribution (see Section V), which is characterized by a higher variance, where most of the packets get out quickly, whereas some of them spend more time in service. Finally, we consider the case of a slow controller where most of the packets spend too much time in service. We model this scenario with exponentially distributed service time (see Section VI).

D. Processes of Interest

Given that we assume non-exponential service times at the data plane and general distribution at the controller, the queueing network at hand is not a Jackson network. Therefore, we cannot exploit the well known results from there.

The metric of interest in this paper is the sojourn time distribution. In order to get valuable insights on that metric, we need to consider the "internal" processes that compose the sojourn time. These processes, also shown in Fig. 1, are

- Inter-arrival time at the data plane (DIAT),
- Inter-departure time from the data plane (DIDT),
- Inter-arrival time at the controller (CIAT),
- Inter-departure from the controller (CIDT).

When we mention these acronyms from now on, we refer to the corresponding probability density function (pdf).

IV. Fast deterministic controller

In the first case, a fast controller with deterministic service times is assumed in combination with the data plane with normally distributed service times. This models the case when the control plane is faster than the data plane. The faster the controller compared to the data plane is, the better the impact

of the controller can be captured by a deterministic function. Results for that system are obtained by simulations.

The mentioned model has been implemented in MATLAB as a packet-based simulation. As already mentioned in Section III, the service times of the data plane are normally distributed. The mean and the standard deviation values are those of T4P4S [7], a P4 software switch, from the measurements in [3], running VxLAN as program and can be seen in Table I.

The controller service times are deterministic. The corresponding parameters for the fast controller are depicted in Table I. The service time S_C has been arbitrarily chosen as in [14] for comparison with the medium-rate controller, which is one order of magnitude slower. Therefore, in this case, the control plane is faster than the data plane. Note that as service times are deterministic, the standard deviation is 0.

We vary the feedback probability p_C and the external arrival rate λ_{ext} in the simulation. In the first case, the external arrival rate is chosen to be very small compared to the data plane service rate (low λ_{ext} case), in the second case it is almost equal to the data plane service rate (high λ_{ext} case), leading to a system utilization close to 1. If the corresponding arrival rate to the controller is equal to or higher than the controller service rate, the external arrival rate is reduced to the highest possible one in order to have a stable queueing system.

The more λ_{ext} is further away from the low and the high regimes, the more the distributions of interest tend to be a combination of those two cases (low and high λ_{ext}). Therefore, we do not consider the medium λ_{ext} scenario here.

In particular, we set $\lambda_{ext} = 200\ s^{-1}$ for a low external arrival rate which is two orders of magnitude lower than the data plane service rate. However, a slow controller with $\mu_C = 100\ s^{-1}$ cannot handle that arrival rate for $p_C = 0.5$ and $p_C = 1$. In these cases, the external arrival rate is reduced to $\lambda_{ext} = 150\ s^{-1}$ and $\lambda_{ext} = 90\ s^{-1}$, respectively, in order to have a stable queue at the control plane. The high arrival rate now depends on the controller service rate and the probability p_C which plays a significant role because the data plane arrival rate depends on the external arrival rate and the controller feedback rate. As the fast controller is faster than the data plane, the latter constrains the highest possible external arrival rate. In order to achieve a high data plane utilization, the arrival rates are set to $\lambda_{ext} = 18500\ s^{-1}$, $\lambda_{ext} = 14000\ s^{-1}$ and $\lambda_{ext} = 10500\ s^{-1}$ for $p_C = 0.1$, $p_C = 0.5$ and $p_C = 1$, respectively. The value of $\lambda_{ext} = 18500\ s^{-1}$ is also chosen for the medium-rate controller and $p_C = 0.1$. However, for $p_C = 0.5$ and $p_C = 1$, the external arrival rate has to be adjusted to the control plane service rate $\mu_C = 4167\, s^{-1}$, i.e., $\lambda_{ext} = 7600\ s^{-1}$ and $\lambda_{ext} = 4000\ s^{-1}$, so that the controller does not get overloaded. In those two cases, the control plane is heavily loaded but not the data plane anymore. Finally, the highest possible arrival rates to the slow controller are $\lambda_{ext} = 999\ s^{-1}$, $\lambda_{ext} = 199\ s^{-1}$ and $\lambda_{ext} = 99\ s^{-1}$ for $p_C = 0.1$, $p_C = 0.5$ and $p_C = 1$, respectively.

TABLE I: Data plane and control plane parameters

Plane	Mean service time	Standard deviation
Data Plane	$45.9\ \mu s$	$4.6\ \mu s$
Fast Controller	$32\ \mu s$	$0\ \mu s$
Medium-rate Controller	$240\ \mu s$	$960\ \mu s$
Slow Controller	$10\ ms$	$10\ ms$

For both arrival rate regions, we consider the system behavior for the probability p_C taking values from the set $\{0, 0.1, 0.5, 1\}$. Higher p_C means increased interaction with the controller (more packets are sent to the controller). All cases run for $100,000$ packets. The packet-related distributions for all relevant processes, depicted in Fig. 1, as well as the sojourn time distributions are directly obtained from simulations. Due to space limitations, only the figures of the most relevant results are shown for each case in this paper. Nevertheless, we provide more results in a tech report [15].

A. Low external arrival rate

In this scenario, λ_{ext} is very low compared to the service rates of the data plane and controller. With no controller, i.e., $p_C = 0$, only DIAT and DIDT exist. Thus, the system behaves like a simple M/G/1 queue. As the external arrival process is a Poisson process, DIAT is exponentially distributed. If the arrival rate is low, DIDT has an exponential-like shape. This behavior can be explained as follows.

When the data plane operates in the region of low utilization (low λ_{ext} compared to its service rate), and the probability of a packet to follow the feedback loop is low (small p_C), the overall arrival process to the data plane can be considered to be Poisson, as most of the packets are external, following a Poisson process. Furthermore, the queue is frequently empty, which implies that the departing packet leaves no packets queued behind. Therefore, given that the inter-arrival time is exponentially distributed and the service time is Gaussian, the inter-departure time is the sum of the exponential and normal distribution, implying that DIDT is an exponentially-modified Gaussian distribution [16], known also as exponormal distribution, defined as

$$f(x) = \frac{\lambda_{ext}}{2} e^{\frac{\lambda_{ext}}{2}(2\mu_D + \lambda_{ext}\sigma_D^2 - 2x)} \operatorname{erfc}\left(\frac{\mu_D + \lambda_{ext}\sigma_D^2 - x}{\sigma_D\sqrt{2}}\right), \quad (3)$$

where μ_D is the service rate of the data plane, and erfc is the complementary error function [16]. As μ_D and σ_D are very small in our considered cases and λ_{ext} is also low, the following approximation holds:

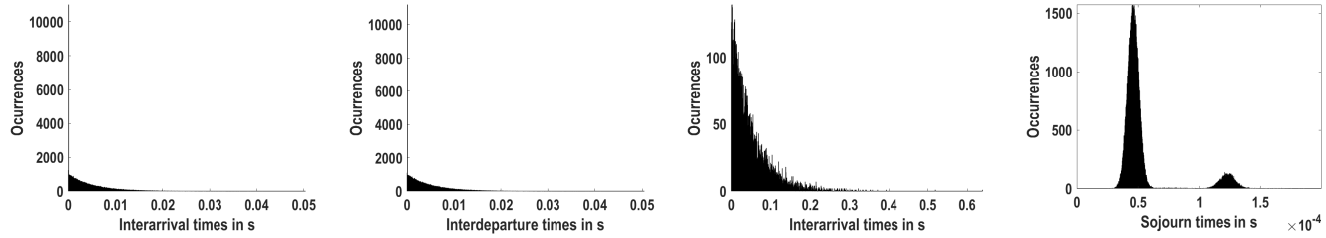
$$\operatorname{erfc}\left(\frac{\mu_D + \lambda_{ext}\sigma_D^2 - x}{\sigma_D\sqrt{2}}\right) \approx 2. \quad (4)$$

Therefore, the pdf of DIDT can be approximated by

$$f(x) = \lambda_{ext} e^{-\lambda_{ext}(x - \mu_D - \frac{1}{2}\lambda_{ext}\sigma_D^2)}, \quad (5)$$

which explains the exponential-alike nature of the inter-departure time distribution in those cases.

When $p_C = 0$, the packet sojourn times are normally distributed around the data plane service time. Packets do not experience queueing due to the low arrival rate and therefore their sojourn times solely depend on the service times. As no controller is involved for $p_C = 0$, these results hold for any



(a) DIAT (b) DIDT (c) CIAT (d) Sojourn times

Fig. 2: Distributions of interest for the fast deterministic controller with a low external arrival rate and $p_C = 0.1$.

control plane service time distributions and this scenario is therefore not considered in the remainder of this work.

Fig. 2 shows the distributions of interest for $p_C = 0.1$. Due to low λ_{ext}, there is no queueing. CIAT depends on the external distribution. According to the theoretical explanation above, CIAT also looks exponential-alike, as in Fig. 2c. As the controller service times are deterministic, only a very small delay is added to the incoming distribution, so CIDT is almost identical to CIAT. The same behavior can be observed in the data plane. DIDT (shown in Fig. 2b) is almost identical to DIAT (shown in Fig. 2a). The theoretical explanation for that behavior is provided on the data plane example that follows.

When the arrival rate in the system is not high, the inter-departure time of the data plane underlies approximately the same distribution as the inter-arrival time. To show this, let us denote with $\Delta t = t_{i+1} - t_i$ the inter-arrival time between two successive arrivals at the data plane. The service times for these arrivals are S_i and S_{i+1}, respectively. Therefore, the departure instant for packet i is $T_i = t_i + S_i$, whereas for packet $i+1$ it is $T_{i+1} = t_{i+1} + S_{i+1}$. Having this in mind, for the inter-departure time from the data plane we have

$$\Delta T = T_{i+1} - T_i = \Delta t + S_{i+1} - S_i. \tag{6}$$

As we assume service times are i.i.d. normally distributed, then $S_{i+1} - S_i$ is also normally distributed, i.e., $\mathcal{N}(0, 2\sigma_D^2)$. For a very low σ_D, the normal distribution $\mathcal{N}(0, 2\sigma_D^2)$ resembles to a considerable extent the Dirac delta function $\delta(x)$. As the components in Eq.(6) are mutually independent, the pdf of ΔT can be written as

$$f_{\Delta T}(x) = f_{\Delta t}(x) * \delta(x) = f_{\Delta t}(x), \tag{7}$$

where * denotes the convolution operation, and $\delta(x)$ is the unit element for that operation.

DIAT shows a high spike around the sum of the data plane and control plane service times, representing the feedback packets. The rest looks exponentially as it depends on the external arrivals. The sojourn time distribution in Fig. 2d consists of two normal-looking distributions. One is around the mean of the data plane service time and the other around the summation of two data plane service times and one controller service time. The first represents the sojourn times of packets not visiting the controller, while the second depicts the sojourn times of feedback packets.

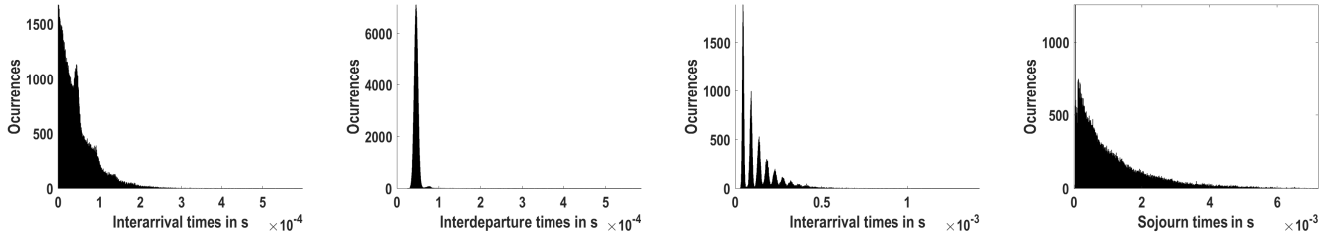
When p_C increases, the impact of feedback packets increases. In particular, the spike in DIAT, and therefore in DIDT increases, while the exponentially-like part decreases. CIAT and CIDT tend to a more expo-normal behavior with a spike in the low inter-arrival times. For the sojourn time, the impact of the second normal distribution increases and of the first one decreases, until it disappears when $p_C = 1$.

B. High external arrival rate

In the region of high λ_{ext}, as the controller is faster than the data plane, the service rate of the latter constrains the external arrival rate in order to have stable queues. Now, queueing occurs on the data plane. When $p_C = 0$, there is no controller interaction and the data plane is heavily loaded. Therefore, DIDT looks normally distributed as the inter-departure times only depend on the service times. The sojourn time distribution is expo-normal, stemming from the exponential inter-arrivals to the queue and the normal service times. These results are not dependent on the controller and will therefore not be mentioned in the remainder of this paper again.

In Fig. 3, several distributions for a high λ_{ext} and $p_C = 0.5$ are presented. On the data plane side, most packets have to be queued, while the control plane is slightly utilized. CIAT (shown in Fig. 3c) now has several normally distributed spikes. They occur on multiples of the service time and are a result of the queueing process. Thus, the time between two packets sent to the controller depends only on the data plane service time and the amount of other packets being served between the two feedback packets. The overall DIDT (see Fig. 3b) is "mostly" normally distributed around the mean data plane service time with an extra component corresponding to packets that do not encounter queueing (i.e., are served as soon as they arrive). The shape of DIAT, depicted in Fig. 3a, is a combination of CIDT, which looks almost like CIAT, and the exponential external arrival distribution. CIDT-related spikes (caused by CIAT) explain the presence of spikes in the overall DIAT. The sojourn time distribution, shown in Fig. 3d, looks expo-normal-like for $p_C = 0$ and high λ_{ext}.

Reducing p_C plummets the impact of feedback packets. For instance, the spikes in DIAT decrease and the overall distribution tends to be exponential-alike. Also, the spikes in CIAT and CIDT become smaller. However, the spike in the sojourn time distribution increases as more packets directly leave the system. DIDT does not change notably. Increasing p_C results in an opposite behavior.



(a) DIAT (b) DIDT (c) CIAT (d) Sojourn times

Fig. 3: Distributions of interest for the fast deterministic controller with a high external arrival rate and $p_C = 0.5$.

V. Medium-rate hyper-exponential controller

The second case is a system consisting of a data plane with normally distributed service times and a control plane with hyper-exponentially distributed service times with two branches. Its pdf is defined as

$$f_{S,C}(x) = p\mu_{C,1}e^{-\mu_{C,1}x} + (1-p)\mu_{C,2}e^{-\mu_{C,2}x}, \quad (8)$$

where p denotes the probability that the packet will experience a small service time, where the latter is exponentially distributed with rate $\mu_{C,1}$. In turn, $1-p$ is the probability that the packet will spend a long time in service. It is also exponentially distributed, but with a much lower rate, i.e., $\mu_{C,2} << \mu_{C,1}$. Most packets experience the very fast service time ($p >> 1-p$). Few packets are processed very slowly, introducing a high variability in the controller service times. This case represents a controller, where many packets are processed very fast and for some packets more complex actions are executed adding a significant delay to the processing time.

We evaluate the performance of this scenario as well. The overall setup is using the same approach, so only the differences are described here. Again, the mean and the standard deviations for the service distributions can be found in Table I. The standard deviation shown for the medium-rate controller belongs to the case of high arrival rate and $p_C = 0.5$. The server characteristics of the controller are described by: $p = 0.9$, $\mu_{C,1} = 100\mu_{C,2}$. The controller mean service time, taken from [14], is higher than data plane mean service time.

A. Low external arrival rate

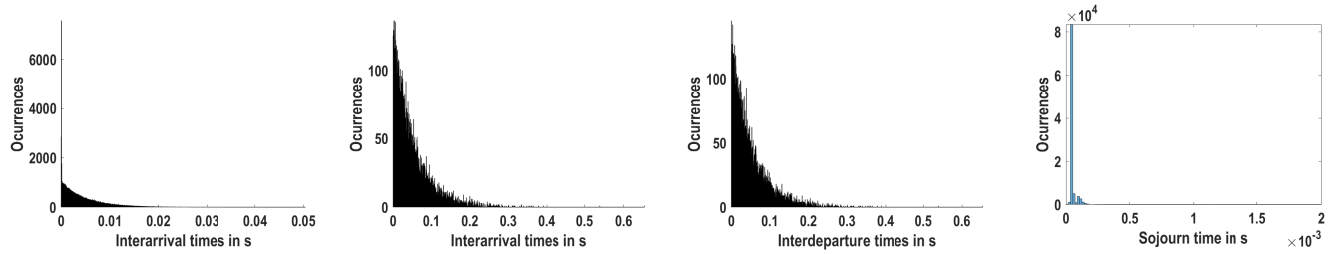
Fig. 4 shows the distributions of interest for a low λ_{ext} and $p_C = 0.1$. Due to the low arrival rate, there is almost no queueing. CIAT in Fig. 4b depends on the external arrivals, whose inter-arrival time is exponentially distributed. Thus, as described in Section IV, CIAT is exponentially-like. At the controller, the packets have a very high probability to experience a very fast service time, faster than the data plane in this case. That explains the spike for small inter-departure times in Fig. 4c compared to CIAT. The overall data plane inter-arrival times in Fig. 4a show a large spike for small times. Fast feedback packets increase the probability of shorter inter-arrival times to the data plane. The remaining part, reminiscent of an exponential distribution, depends on the external arrivals. DIDT follows DIAT, as described in Section IV. The aforementioned distributions exhibit a large variance as some packets are served by the controller with a high service time in contrast to most of the packets served very fast. This can be observed from the sojourn time distribution in Fig. 4d. While the sojourn time of most packets lies around the mean data plane service time, the sojourn time of the remaining few packets causes the high variance in the distribution.

Higher values of p_C increase the impact of the controller and the feedback packets. The spike in DIAT for small inter-arrival times increases and the exponential-like remaining part of the distribution decreases. CIAT follows a more exponormal distribution with a spike at small values. The same behavior can be observed for CIDT. However, the spike of small controller inter-departure times grows larger than for CIAT as most of the feedback packets experience a very fast controller service time. As more packets are sent to the controller, more packets experience long service times at the control plane, resulting in a decrease of the spike in the sojourn times distribution and a higher sojourn times variance.

B. High external arrival rate

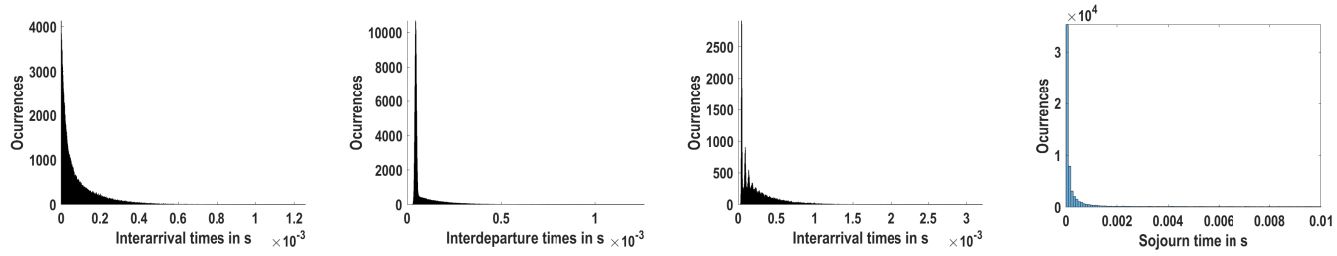
For a high λ_{ext}, there is more queueing on both the data plane and controller. As the service rate of the controller is lower than that of the data plane, the controller is the limiting factor for the arrival rate in order not to get an unstable queue. Fig. 5 shows all distributions of interest for the case when $p_C = 0.5$. CIAT (depicted in Fig. 5c) shows spikes at multiples of the data plane service time. These spikes correspond to the interval between two packets being sent to the controller in the data plane waiting queue. Most of the packets experience a very fast controller processing, so CIDT has a large spike for fast inter-departure times. DIAT (shown in Fig. 5a) is exponentially-like with a higher probability for lower inter-arrival times. DIDT (shown in Fig. 5b) mostly looks normally distributed with some exponentially-like "flavor". This is due to the data plane inter-departure times for queued packets depending on the normally distributed data plane service times. The rest is caused by the time the data plane is idle and waits for packets. The sojourn time distribution shows a large spike around the data plane mean service time. All distributions are subject to a large variance, i.e., the very slow service times of the controller lead to a few very widely spread sojourn times.

Lower p_C result in lower impact of feedback packets. Thus, the spike in DIAT decreases and DIAT tends to look exponential. DIDT exhibits a more normal-alike behavior as



(a) DIAT (b) CIAT (c) CIDT (d) Sojourn times

Fig. 4: Distributions of interest for the medium-rate hyper-exponential controller with a low external arrival rate and $p_C = 0.1$.



(a) DIAT (b) DIDT (c) CIAT (d) Sojourn times

Fig. 5: Distributions of interest for the medium-rate hyper-exponential controller with a high external arrival rate and $p_C = 0.5$.

the load on the data plane gets higher for lower values of p_C. That is due to the controller limiting the external arrival rate for higher values of p_C. CIAT shows more spikes because it is more probable that two consecutive "to-be" feedback packets have more packets in between them in the data plane waiting queue. As fewer packets go to the controller, also the spike in CIDT decreases. The overall sojourn time in Fig. 5d tends to have a much heavier tail than the exponential distribution.

VI. Slow exponential controller

In the third case, controllers with a slower service rate than the data plane, such as ONOS [17] controller, are considered. The mean sojourn time of the ONOS controller was measured to be $8\,ms$ [13], significantly lower than the T4P4S VxLAN data plane (see Table I). This system is again evaluated with simulations. The mean value of the exponentially distributed service time at the controller is $10\,ms$ (Table I).

A. Low external arrival rate

Fig. 6 shows the distributions of interest for a low λ_{ext} and $p_C = 0.5$. CIAT, shown in Fig. 6c, looks exponentially-like. This behavior for a low data plane utilization is explained in Section IV. As the controller service times are also exponentially distributed, DIAT in Fig. 6a, consisting of the exponentially distributed service times and the exponentially-like feedback distribution also looks similar to an exponential distribution. DIDT, shown in Fig. 6b, follows DIAT, as already explained in Section IV. Thus, all distributions for such a system with a slow exponential controller are exponential-like. The distribution of the packet sojourn times in Fig. 6d shows a spike for fast sojourn times, representing the sojourn times of packets directly leaving the system. The other times are influenced by controller processing and queueing. Changing the value of p_C does not change the exponentially-like behavior of the distributions. Moreover, for $p_C = 1$ the sojourn time distribution also looks exponentially (it is mostly influenced by the slow controller). As the data plane is much faster than the control plane, it can be approximated by a constant (deterministic) value only slightly shifting the exponormal distribution coming from the controller. Therefore, as all packets are sent to the controller, the sojourn time is exponentially-like, similar to CIAT.

B. High external arrival rate

Considering high λ_{ext} in the system with a slow controller corresponds actually to the low λ_{ext} case. As the controller is much slower than the data plane, the highest possible external arrival rate is limited by the controller. Even this highest arrival rate is still very small compared to the data plane service rate. Thus, the results of Section VI-A apply here as well.

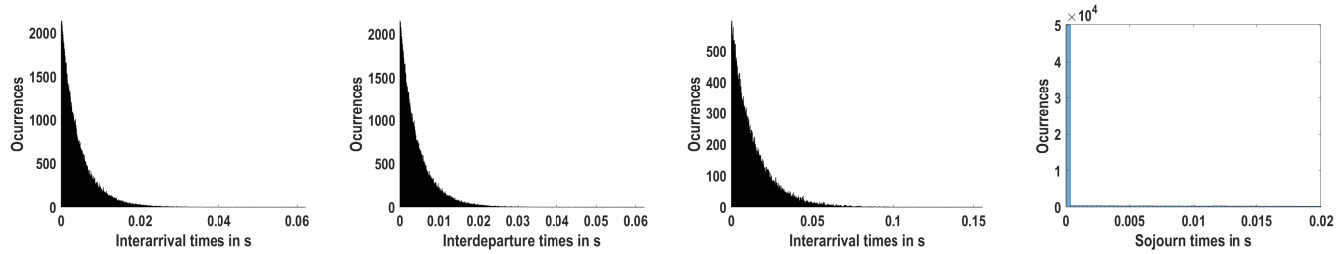
VII. Performance comparison

As the sojourn time is the metric of interest in this work, which is important to better understand and predict the behavior of a P4 switch, in this section we analyze and compare the *coefficients of variation* of the sojourn time for the three controllers. Besides obtaining the first moments of the sojourn and discussing about the qualitative behavior of the sojourn time distribution, quantifying its dispersion around the mean is also very important. The coefficient of variation of random variable X is defined as

$$c_V = \frac{\sigma_X}{\mathbb{E}[X]}, \tag{9}$$

i.e., as the ratio of the standard deviation and the mean.

In Fig. 7, the three different controllers are compared for a low λ_{ext}. The fast deterministic controller already shows very low values of c_V. For $p_C = 0.1$ and $p_C = 0.5$



(a) DIAT (b) DIDT (c) CIAT (d) Sojourn times

Fig. 6: Distributions of interest for the slow exponential controller with a low external arrival rate and $p_C = 0.5$.

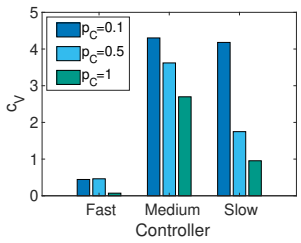


Fig. 7: The sojourn times c_V for low arrival rate.

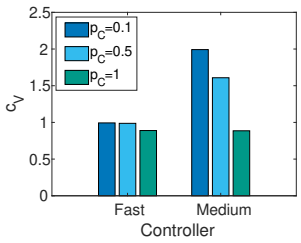


Fig. 8: The sojourn times c_V for high arrival rate.

they are 0.44 and 0.46, respectively. In this system without queueing, packets either leave the system immediately or visit the controller. As both are very fast (the data plane and the controller) with low standard deviations, where for the controller it is actually 0, the dispersion they introduce to the sojourn time is low. Thus, the values of c_V in those cases are very small. Increasing the p_C to 1 results in a c_V drop to 0.071. Each packet now visits the controller, which means they experience the normally distributed data plane twice and the deterministic control plane once. As the normal data plane is so fast with a low variance, the overall packet sojourn times are very predictable, i.e., do not vary much.

In the case of the medium-rate hyper-exponential controller, the values of c_V are 4.3, 3.62 and 2.7 for increasing values of p_C. Compared to the fast deterministic controller case, the c_V is higher due to the hyper-exponential distribution having a higher variance. The lowest value of p_C corresponds to the highest value of c_V due to the impact of the very high controller service times for a small number of packets and the majority of the packets leaving the system immediately after being processed by the very fast data plane. For increased values of p_C, the latter packets are less and those more that are served by the controller very fast. Thus, the overall behavior is more predictable, resulting in lower values of c_V. However, due to the high variance of the hyper-exp. controller, the values of c_V are still high compared to other controllers.

A similar behavior can be observed for the slow exponential controller. For $p_C = 0.1$, we obtain $c_V = 4.18$, which is comparable to the hyper-exponential case. The reason for that lies in the huge difference in the sojourn time for packets leaving the system immediately and those being sent to the controller. Whereas the former are processed very fast in the data plane, the latter have a very high sojourn time due to the slow controller. For $p_C = 0.5$, c_V drops to 1.75 as more packets are now visiting the controller. When all packets visit the controller, $c_V = 0.95$, which is close to the c_V of an exponential distribution. In this case, the slow exponential controller has the largest impact on the sojourn time.

Fig. 8 depicts the values of c_V for a high λ_{ext} for the fast and medium-rate controllers. The slow controller is not considered as its low service rates limit the highest possible arrival rate. This highest possible arrival rate is very low and therefore the results for the low arrival case apply. For the fast deterministic controller, the values of c_V are 0.99, 0.99 and 0.89, which are very close to the exp. distribution. This is due to the behavior of the sojourn times of a G/G/1 queue tending to an exp. distribution in the very high utilization regime [10]. In this case, the data plane is heavily loaded.

The values of c_V for the medium-rate hyper-exp. controller are 1.99, 1.6 and 0.89. For $p_C = 0.1$, only a small fraction of packets is forwarded to the controller and hence the data plane can be heavily loaded without leading to an unstable control plane. The slowly served feedback packets compared to the very fast packets directly leaving the system yield a large variance in the sojourn times and hence, to a higher c_V. For $p_C = 0.5$ and $p_C = 1$, the behavior of the system changes. The data plane cannot get overloaded anymore without overloading the controller. So, the controller now limits the external arrival rate and most packets at the controller encounter queueing. As data plane is not heavily loaded anymore, for those values of p_C the c_V drops. When all packets visit the controller, $c_V < 1$. The reason lies in the fact that all packets in this case traverse the same path (twice the data plane and once the controller), where the control plane is heavily loaded and data plane is not. For the former, we mimic the time the packet spends there with an exponential distribution (high utilization regime), whereas in the latter, given that the service time has low variability, the queueing adds only slightly to the c_V (but keeping it < 1). Combining both effects then results in a total $c_V < 1$.

VIII. Related Work

Modeling the performance of P4-based devices has been studied very extensively in the last few years, but in all the cases the analysis is confined to obtaining first-order statistics. To the best of our knowledge, there are no other works that reveal to the full extent (in terms of the distribution) the packet behavior in a P4-based system. At the data plane level, the

authors in [18] proposed a benchmarking suite for evaluating different metrics of different P4 targets. They evaluated the forwarding latency of different software-based devices when running different elementary P4-14 operations.

The authors in [3] went one step further by evaluating the delay of running different P4-16 operations on different P4 hardware and software targets. They used the conducted measurements to propose models for estimating the packet forwarding latency when running arbitrary P4 programs on any P4 target's data plane. However, the problem of inferring the distributions of the metrics of interest has not been tackled in any of these works, and obtaining the first-order related metrics in some cases will simply not be sufficient. On the other hand, here we analyze the P4-related system and look in more depth at its operation under different controller behaviors.

Further, the authors in [19] model and evaluate the key properties of P4 targets' data planes, where these characteristics vary depending on the processing platforms. At the controller, [13] introduces a new benchmarking tool for evaluating P4Runtime-based SDN controllers, and presents evaluation results of the ONOS controller, collected using that tool. Predicting the worst-case latency that can be encountered in SDNs and P4-based data planes has been the focus of [20], where network calculus is used to model the behavior. However, that approach can be quite restrictive as we do not necessarily face the worst possible scenario. Furthermore, the worst possible scenario can be obtained as a special case using our approach too (when the external arrival is close to the boundary of the operational region of either the data plane or the controller).

In terms of the model used, the queueing networks with feedback have already been analyzed in [5], [21], [22]. However, common to [5], [21], and [22] is the fact that in their queueing network with feedback the packet can go multiple times through the feedback branch (corresponding to the control plane in our case). This is different from our setup as we assume that the packet can go back to the controller at most once. Hence, these models are not suitable for our scenario. Moreover, in [5], [21], and [22] there is no discussion provided about the distribution of the sojourn time in the system.

The work most related to ours is [4], which models the performance of a complete P4-based system, with data and control planes, using a queueing-based network model. However, only the average sojourn time is derived in [4] for the special case of exponentially distributed service times in the data plane and controller, and an approximation is provided for other distributions. Furthermore, in [4] there are no indications as to the behavior of the distribution of the sojourn time, and in particular of its second moment, as a function of the variability of the service times in both the data plane and the controller.

IX. CONCLUSION

In this paper, we analyzed the behavior of P4-supported data plane in an SDN, where the controller activity undergoes different regimes of operation. We modeled the system with a queueing network with feedback. Then, we argued why the service time in the data plane can be approximated with a normal distribution, and provided insights on the distribution of the metrics of interest that can be encountered in different operation regimes. We also showed that when more packets are forwarded to the controller, the variability of the packet sojourn time in such a system decreases for medium-rate and slow controllers. Additionally, when increasing the arrival rate to the system, the variability of the sojourn times for the fast deterministic controller increases, whereas the opposite effect is observed for medium-rate hyper-exponential controller. In the future, we plan to develop suitable analytical models that will enable to obtain closed-form approximations for higher moments of the sojourn time.

REFERENCES

[1] 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, 2008.

[2] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014.

[3] H. Harkous, M. Jarschel, M. He, R. Pries, and W. Kellerer, "P8: P4 with predictable packet processing performance," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, 2021.

[4] H. Harkous, N. Kröger, M. Jarschel, R. Pries, and W. Keller, "Modeling and performance analysis of P4 programmable devices," in *Proc. of IEEE NFV-SDN*, 2021.

[5] W. Whitt, "The queueing network analyzer," *The Bell System Technical Journal*, vol. 62, no. 9, 1983.

[6] P. L. Consortium. (2022) Behavorial model (bmv2). [Online]. Available: https://github.com/p4lang/behavioral-model

[7] P. Vörös, D. Horpácsi, R. Kitlei, D. Leskó, M. Tejfel, and S. Laki, "T4P4S: A target-independent compiler for protocol-independent packet processors," in *Proc. of IEEE HPSR*, 2018.

[8] Netronome. (2022) Netronome agilio smartnics. [Online]. Available: https://www.netronome.com/products/smartnic/overview/

[9] N. Zilberman, Y. Audzevich, G. A. Covington, and A. W. Moore, "Netfpga sume: Toward 100 Gbps as research commodity," *IEEE Micro*, vol. 34, no. 5, 2014.

[10] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*, 4th ed. Wiley-Interscience, 2008.

[11] P. Billingsley, *Probability and Measure*, 3rd ed. Wiley, 1996.

[12] S. M. Ross, *Stochastic Processes*. John Wiley & Sons, 1996.

[13] H. Harkous, K. Sherkawi, M. Jarschel, R. Pries, M. He, and W. Kellerer, "P4RCProbe for evaluating the performance of P4Runtime-based controllers," in *Proc. of IEEE NFV-SDN*, 2021.

[14] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," in *Proc. of ITC*, 2011.

[15] N. Kröger, H. Harkous, F. Mehmeti, and W. Kellerer, "Looking beyond the first moment in P4 systems with controller feedback," https://sites.google.com/view/fidanmehmeti, 2022, tech report.

[16] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Wiley, 2010.

[17] *ONOS-Open Network Operating System*, Open Networking Foundation.

[18] H. T. Dang, H. Wang, T. Jepsen, G. Brebner, C. Kim, J. Rexford, R. Soulé, and H. Weatherspoon, "Whippersnapper: A P4 language benchmark suite," in *Proc. of SoSR*, 2017.

[19] D. Scholz, H. Stubbe, S. Gallenmüller, and G. Carle, "Key properties of programmable data plane targets," in *Proc. of ITC*, 2020.

[20] M. Helm, H. Stubbe, D. Scholz, B. Jaeger, S. Gallenmüller, N. Deric, E. Goshi, H. Harkous, Z. Zhou, W. Kellerer, and G. Carle, "Application of network calculus models on programmable device behavior," in *Proc. of ITC*, 2021.

[21] J. M. Harrison and V. Nguyen, "The QNET method for two-moment analysis of open queueing networks," *Queueing Systems*, no. 6, 1990.

[22] B. R. Haverkort, "Approximate analysis of networks of PH—PH—1—K queues: Theory & tool support," in *Quantitative Evaluation of Computing and Communication Systems*. Springer, 1995.