

Analytical Model for SDN Signaling Traffic and Flow Table Occupancy and Its Application for Various Types of Traffic

Christopher Metter, Michael Seufert, Florian Wamser, Thomas Zinner, and Phuoc Tran-Gia

Abstract—Software defined networking (SDN) has emerged as a promising networking paradigm overcoming various drawbacks of current communication networks. The control and data plane of switching devices is decoupled and control functions are centralized at the network controller. In SDN, each new flow introduces additional signaling traffic between the switch and the controller. Based on this traffic, rules are created in the flow table of the switch, which specify the forwarding behavior. To avoid table overflows, unused entries are removed after a predefined time-out period. Given a specific traffic mix, the choice of this time-out period affects the tradeoff between *signaling rate* and *table occupancy*. As a result, network operators have to adjust this parameter to enable a smooth and efficient network operation. Due to the complexity of this problem caused by the various traffic flows in a network, a suitable abstraction is necessary in order to derive valid parameter values in time. The contribution of this paper is threefold. First, we formulate a simple analytical model that allows optimizing the network performance with respect to the table occupancy and the signaling rate. Second, we validate the model by means of simulation. Third, we illustrate the impact of the time-out period on the signaling traffic and the flow table occupancy for different data-plane traffic mixes and characteristics. This includes scenarios with single application instances, as well as multiple application instances of different application types in an SDN-enabled network.

I. INTRODUCTION

SOFTWARE Defined Networking (SDN) is a new networking paradigm overcoming various drawbacks of current communication networks. The control and data plane of switching devices is decoupled and all control functions are centralized within the network controller(s). With this simple principle, new networking scenarios are possible, facilitated by the introduction of increased control capabilities, dynamic modification of networking parameters, or the control of traffic flows on different granularities. Further, the applicability of these features is not restricted to a specific type of network,

but can be used in access networks, data center networks, and wide area networks [1]–[3].

From a technical perspective, currently, two modes of operation are discussed for SDN-enabled networks: proactive and reactive forwarding mode. In proactive mode, most of the traffic traversing the topology is assumed to be known, e.g., in a data center. Traffic flow rules can be pre-installed in the network. Reactive mode, in contrast, is ideal for highly dynamic traffic where no information on the traffic mix is previously known. In such a scenario, SDN rules are pushed to the switch based on incoming flows on the data plane. Typically, flow rules are then provided in a network-wide manner, i.e., controller traffic is not induced at every switch, but only at the first one. Thus, each new arriving flow at the switch triggers a signaling request to the controller. The controller defines a route through the topology and installs the new flow in the devices of the topology. In SDN, in general, multiple controllers per switch, with various controller architectures, are possible. Currently, multi-controller architectures are mostly used in order to increase the availability and reliability of the control plane. In order to reduce the possibility of contradicting commands sent from controllers to the switch, the OpenFlow protocol supports, for a example, a master- and slave role assignment of the connected controllers. Thus, a switch might have connections to multiple controllers with each of the connected controllers receiving updates from the data plane, but, only the master controller is allowed to install new rules. In case a controller fails, the ownership of the affected switches is transferred to a controller which was formerly in slave role. In this paper we will only work with one single controller instance. Nevertheless, the findings in this paper are also applicable for multi-controller environments. Depending on the chosen controller architecture, only the signaling load on side of the switch increases, as multiple controllers need to be informed about changes.

Since the flow table space is limited in terms of TCAM (ternary content-addressable memory) and CAM (content-addressable memory), unused flow rules should be removed. This is done with the help of a *time-out value* which can be configured at installation time of the flow. A flow is removed exactly when the inter-arrival times of packets within the flow exceed the time-out value. In case of additional new packet arrivals of this flow, the controller is then again involved in the forwarding decision resulting in additional control plane traffic and waiting times for the data plane traffic.

(Corresponding author: Christopher Metter.)

The authors are with the University of Würzburg, 97074 Würzburg, Germany (e-mail: christopher.metter@informatik.uni-wuerzburg.de; seufert@informatik.uni-wuerzburg.de; wamser@informatik.uni-wuerzburg.de; zinner@informatik.uni-wuerzburg.de; trangia@informatik.uni-wuerzburg.de).

Digital Object Identifier 10.1109/TNSM.2017.2714758

Given a specific traffic mix, the choice of the time-out period affects the trade-off between signaling rate and table occupancy. On the one hand, small time-outs might result in unnecessary signaling traffic, control plane overload, and long response times for control actions [4]. On the other hand, long time-outs might cause a high flow table occupancy and table overflows. As a result, providers have to adjust this parameter to enable a smooth and efficient network operation. The overall complexity of this problem including uncertainty of traffic mix and application characteristics does not allow a one-time optimization of the time-out period. Instead, the relevant system parameters have to be monitored in regular intervals and the time-out parameter has to be adjusted if necessary. To derive viable parameter settings in a timely manner, an appropriate abstraction is required.

In this paper, an analytical model is presented which can be used to analyze the outlined key performance parameters of SDN in reactive mode. It is the continuation of the work in [5], where the model was initially presented and described. Based on various traffic mixes and applications, different scenarios in the reactive forwarding mode of the controller are investigated. With the model we study the control plane traffic between the controller and switch. Following the same principle, we also derive the utilization of the flow table in the switch.

The contribution of this paper is threefold. Firstly, we provide the analytical model as a tool ready to use for a network operator to analyze control plane traffic and flow table utilization. Secondly, the model is validated using a discrete-event simulation. Thirdly, we show the impact of traffic flow characteristics on control plane traffic for different scenarios.

The remainder of the paper is structured as follows. In Section II, related work on SDN and network performance analysis is discussed. Section III presents the analytical SDN model as well as the used methodology. In Section IV, we evaluate the impact of different TCP flow characteristics and time-outs on the controller traffic and the flow table occupancy in a realistic scenario. Section V shows the simulation for validating the model. It is applied together with the model for an extended scenario, where several applications with different characteristics are active simultaneously in the network. Finally, Section VI concludes the paper.

II. RELATED WORK

This section features work and research with the focus on modeling switch-controller traffic and/or the impact of different time-out values.

Jarschel *et al.* [6] modeled the basic OpenFlow switch model based on M/M/1-S queues for switch, controller, and the interaction between these two elements, in order to analyze forwarding speed and blocking probabilities. Their results indicate, that the packet sojourn time in an OpenFlow-enabled network is mainly dependent on the controller. Mahmood *et al.* [7] extended this work, as it is only viable for one single forwarding element and lacks correctness for highly bursty network traffic. Jarschel *et al.* [6] enhance the

model of to the case of multiple nodes by using an Open Jackson network in [8]. The presented model is used to evaluate an SDN systems average packet sojourn time. Afterwards, they validate their model by simulation for the case of serial topologies. Additionally, the impact of bursty traffic on the models output can be analyzed. Reference [9] created an OpenFlow-based queuing model that provides the average packet sojourn time through a switch in large-scale OpenFlow networks. Their numerical analysis concludes that the packet sojourn time mainly depends on the packet processing capability of the controller. To demonstrate the correctness of their model, multiple measurements matching the results have been conducted. Azodolmolky *et al.* [10] present an analytical model to investigate the impact of composed application traffic on the interaction between switch and controller by using network calculus. The presented switch model captures the packet delay and the buffer length inside the SDN switch according to the parameters of a cumulative arrival process. In [11] an analytical network calculus model focusing on an upper bound for the processing delay of unknown flows arriving at a switch is created. The results of the model are validated by simulation. According to Lin *et al.* [12] their model presents a faster way to predict the performance of SDN for any type of SDN deployment than benchmarking tools. In their paper, they utilize stochastic network calculus methods to analyze and evaluate the performance of an SDN deployment. To validate their results, they ran simulations and measurements in a testbed. Miao *et al.* [13] introduce an analytical model for the SDN architecture by using Markov-Modulated Poisson Process arrivals. According to their results, key performance indicators such as average latency and network throughput can be predicted by their model. An extensive OMNeT++ simulation experiment approves their results.

Beigi-Mohammadi *et al.* [14] propose a model to study the efficiency and scalability of an application-aware software-defined infrastructure. After presenting the model, the authors conduct testbed measurements to verify the validity of their work. While being able to identify a bottleneck in the scalability of their cloud testbed, the authors do not touch the scalability of the used flow table space.

Analyzing different mechanisms of flow table updates, Liu *et al.* [15] study the impact of bandwidth and flow table size on the performance of flow updates. Both, a qualitative and quantitative analysis of these trade-offs in multiple realistic network topologies are presented. In contrast to our work, the authors focus on the updates of already existing flows within a flow table. Therefore, possible flow table space problems are not considered.

All these papers did not analyze neither flow time-out, nor table occupancy. The impact of flow table time-out length on performance and table occupancy through measurements is analyzed in [16]. Additionally, multiple caching algorithms for a flow table are compared and evaluated. Their results indicate that, with an increasing time-out, the probability of an arriving packet triggering a request to the controller decreases exponentially, whilst the table size grows linearly. Depending on the characteristics of the data-plane traffic, the authors

are also able to identify good starting points for the time-out value: 5 and 10 seconds. These values, though, are not put in relation to the characteristics of the analyzed traffic. Based on their observations, the authors propagate a dynamically chosen time-out value. According to [17], one of the main scalability problems of SDN controllers is that the controller is often simply overwhelmed by the number of requests. To overcome this issue, the authors propose to adjust flow time-outs based on the mean inter-arrival time of packets per flow. Their results indicate that the dynamic modification of the time-outs, in dependence of the quality of their prediction, may decrease the controller load by almost 10%. Zhu *et al.* [18] point out the importance of suitable time-outs for each flow as well as a load awareness of the flow table. They propose a mechanism to assign flow time-outs according to flow characteristics. Additionally, a feedback control to dynamically adjust the max time-out value according to the current load of the flow table is presented. Kim *et al.* [19] choose an LRU caching algorithm to reduce the table-miss rate of the switch. Thus, the controller load can be reduced. In [20] rule-caching is also used to increase the flow table-hits. Their design is based on four criteria: elasticity, transparency, fine-grained, and adaptability, and satisfies their requirements. Reference [21] describe important performance characteristics of flow-tables from different manufactures by measurements. Their goal is to make controllers use of flow-tables more efficient. The main outcome of this work is that OpenFlow switches, although implementing the same OpenFlow protocol version, differ widely.

Several important work has been done in the area of modeling network traffic, which lays the foundations for analyzing the impact of flow-entry time-out on the overall performance in SDN. Ciucu and Schmitt [22] present an overview on the general usability and applicability of network calculus for modeling the performance of networks. In 1998 and 2000, Feldmann *et al.* [23], [24] presented fundamental work for modeling WAN traffic. The approach we are presenting in the next chapter features a universal analysis and is easily modifiable to a custom architecture. In order to imitate different general arrival processes, we adopt a two-moment substitution as proposed in [25], using Markovian arrival processes. In contrast to the Markov property, it has been shown that there is a long-range dependency in network traffic, as noted in [26]. Andersen and Nielsen [27] created a model to represent these findings in superpositions of two-state Markov-modulated Poisson processes. Duffield and Whitt [28] and Sriram and Whitt [29] also present a candidate for source traffic models.

In [5] we introduced our model to analyze the impact of the flow table time-out value T_0 on the controller signaling rate of multiple applications, i.e., how many requests per second are triggered by an application, and the overall switch table occupancy, i.e., what is the percentage of time a flow is actually stored in the flow table. As shown in our results, the model produces valid output for the case of one single application generating one or multiple flows at the same time. Now we want to discuss the validity of our model by presenting accompanying simulations.

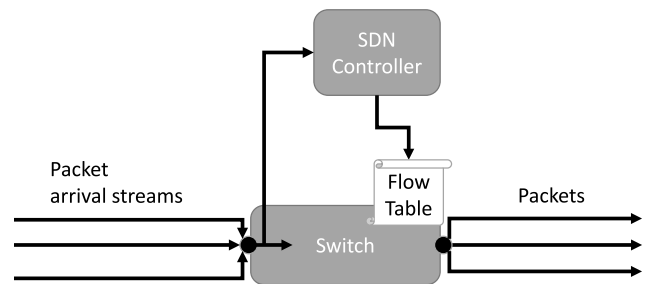


Fig. 1. SDN switching model.

III. MODELING CONCEPT, METHODOLOGY AND ANALYSIS

A. Scenario Description

For this analysis, we consider a single SDN switch, which is connected to a reactive SDN controller, see Figure 1. Multiple TCP flows are active in the network, thus, packet streams arrive at the SDN switch and have to be forwarded. The presented model can be applied for any flow-based traffic with known packet-inter-arrival time. In this paper we focus on TCP flows.

At the start of operation the SDN switch has no knowledge on how to handle any arriving packet. When the first packet of a TCP flow arrives, the SDN switch produces a flow table miss and sends a request on how to handle the new flow (`Packet_In` message) to the SDN controller. The SDN controller replies with a flow rule, which is stored in the flow table of the SDN switch. Any successive packet belonging to the same flow can now be processed by the switch independently. Due to the limitations in flow table size, flow rules cannot be kept forever in the switch. Therefore, current implementations of SDN switches discard entries after these entries have been rendered useless.

Therefore, the arrival of a packet starts a time-out period T_0 for the given flow rule. If the next packet of the flow arrives before T_0 , the time-out period is restarted. If no packet arrives within T_0 , the flow rule is discarded by the switch. If another packet of this flow arrives after T_0 , the packet will cause a flow table miss, and thus, the described procedure repeats. T_0 can be set to an arbitrary time-out value between 0 and `Integer max`. A value of 0 indicates an infinite idle time-out (no idle time-out condition), any other value a time-out value in seconds [30].

B. Single Flow Model

First, we investigate the situation in which a packet stream of a single TCP flow arrives at the SDN switch. Figure 2 illustrates the situation and introduces the used variables. We assume that the inter-arrival times of the packets of the TCP flow follow a general independent distribution A with $A(t) = P(A \leq t)$. We divide the TCP flow into subflows, which are characterized as the time periods, from the setting of the flow rule to its time-out to the subsequent setting of the flow rule. This means, based on the time-out T_0 , the subflow contains packets with inter-arrival times smaller than T_0 until the time-out of the flow rule, and continues until the next packet starts

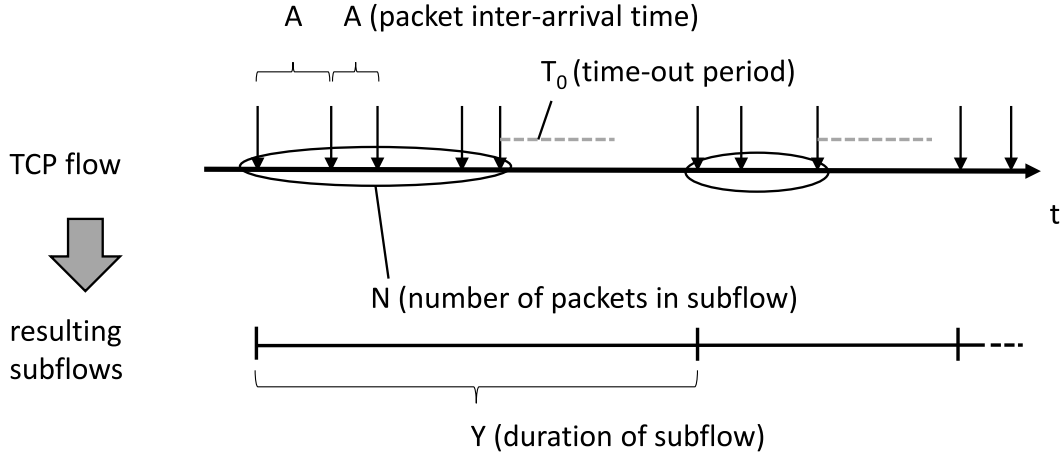


Fig. 2. Illustration of resulting subflows and their characteristics, i.e., number of packets N and duration Y , based on the inter-arrival time distribution A of the TCP flow and the time-out period T_0 of the SDN switch.

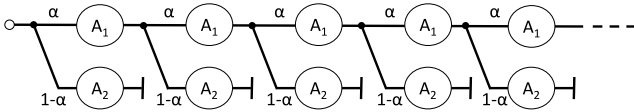


Fig. 3. Phase diagram for the composition of Y .

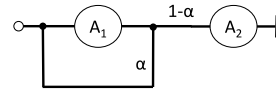


Fig. 4. Feedback loop for subflow duration Y .

a new subflow. The following parameters will be used and introduced in the following sections:

- N : The random variable of the number of packets in a subflow
- Y : The random variable of the duration of a subflow (in seconds)
- η_Y : The signaling rate, i.e., the inter-arrival time of requests towards the controller (in 1/seconds)
- ρ_Y : The table occupancy, i.e., the percentage of time a flow rule is present in the flow table.

As a subflow starts with one packet and every subsequent packet belongs to the same subflow if it arrives within the time-out period T_0 , and else the subflow ends, the number of packets in a subflow N follows a geometric distribution. For shortening reasons, we introduce α as the probability that the inter-arrival time is less or equal than T_0 , i.e., $\alpha := P(A \leq T_0) = A(T_0)$. Throughout this paper, we assume $\alpha < 1$, otherwise the single resulting subflow would be identical to the original flow. Eventually, the distribution of N is given by Equation (1).

$$P(N = k) = \alpha^{k-1} \cdot (1 - \alpha), \quad k \in \{1, 2, \dots\} \quad (1)$$

This insight helps to derive the duration of a subflow Y . We define $A_1 := \frac{A \cdot 1_{\{t \leq T_0\}}}{\alpha}$ as a random variable with the truncated conditional distribution, which gives the inter-arrival time of the packets in case the arrival is less or equal to T_0 . Moreover, we define $A_2 := \frac{A \cdot 1_{\{t > T_0\}}}{1 - \alpha}$ as the corresponding random variable in case the arrival is greater than T_0 . Then, we consider the subflow duration Y as depicted in Figure 3. Y can be iteratively composed of A_1 phases, such that with probability α an A_1 phase is added to Y , until the subflow times out with a phase A_2 with probability $1 - \alpha$. Consequently, the random number of A_1 phases in Y follows the shifted geometric distribution

$N' := N - 1$. Thus, Y can be written as a sum of a random number of random variables:

$$Y = A_{1(1)} + A_{1(2)} + \dots + A_{1(N')} + A_2 \quad (2)$$

Figure 3 can be transformed into a feedback loop as depicted in Figure 4. This can be handled by means of standard control theory, which gives the Laplace transform $\Phi_Y(s)$ as presented in Equation (3).

$$\Phi_Y(s) = \frac{(1 - \alpha) \cdot \Phi_{A_2}(s)}{1 - \alpha \cdot \Phi_{A_1}(s)} \quad (3)$$

We can now compute the moments of Y to obtain the expectation and coefficient of variation of Y in Equation (4) depending on A_1 and A_2 . The signaling rate η_Y , indicating the rate of `Packet_In` messages arriving at the controller, is thus the inverse of the average subflow duration. It can be seen that the characteristics of Y depend on the moments of A_1 and A_2 . Obviously, these moments are influenced by the characteristics of the packet arrival process A of the TCP flow and the threshold T_0 of the switch. We will investigate this relationship in detail in Section IV by utilizing substitute arrival processes, which are introduced in Section IV-A.

$$\begin{aligned} E[Y] &= -\Phi_Y'(0) = \frac{\alpha}{1 - \alpha} E[A_1] + E[A_2] \\ \eta_Y &= \frac{1}{E[Y]} = \frac{1}{\frac{\alpha}{1 - \alpha} E[A_1] + E[A_2]} \\ E[Y^2] &= \Phi_Y''(0) \\ &= \frac{2\alpha^2 E[A_1]^2}{(1 - \alpha)^2} + \frac{2\alpha E[A_1] E[A_2] + \alpha E[A_1^2]}{1 - \alpha} + E[A_2^2] \end{aligned}$$

$$\begin{aligned}
\text{Var}[Y] &= E[Y^2] - E[Y]^2 = \frac{\alpha E[A_1^2] - \alpha^2 \text{Var}[A_1]}{(1 - \alpha)^2} + \text{Var}[A_2] \\
c_Y &= \frac{\sqrt{\text{Var}[Y]}}{E[Y]} \\
&= \frac{\sqrt{\text{Var}[A_2](1 - \alpha)^2 + \alpha E[A_1^2] - \alpha^2 \text{Var}[A_1]}}{\alpha E[A_1] + (1 - \alpha)E[A_2]} \quad (4)
\end{aligned}$$

The removal of each subflow from the switch table after the time-out T_0 has the characteristics of an on-off-process. The on-phase represents the time in which the flow rule is stored in the switch table, and its random variable Y_{on} can be computed by substituting A_2 in the above calculations with the deterministic random variable of the time-out T_0 .¹ The off-phase, being the time in which the flow rule is not stored in the switch table, is given by the random variable $Y_{off} := A_2 - T_0$. Consequently, $Y = Y_{on} + Y_{off}$, and the switch table occupancy ρ_Y for a subflow Y , i.e., the percentage of time a flow rule is present in the flow table, can be computed by Equation (5).

$$\rho_Y = \frac{E[Y_{on}]}{E[Y]} = \frac{\frac{\alpha}{1-\alpha}E[A_1] + T_0}{\frac{\alpha}{1-\alpha}E[A_1] + E[A_2]}. \quad (5)$$

C. Composite Model - The Case With Multiple TCP Flows

After characterizing the subflows of a single TCP flow, we now transfer our findings to the case with multiple TCP flows. Typically, not all users in a network topology are active at the same time. Based on the analysis in the previous subsection about the arrival-rate and the service time of a single user, we now draw conclusions about the number of simultaneous users in a system.

Therefore, we assume a memoryless arrival process of TCP flows with rate λ , each being active for a certain time following a general independent distribution B . Moreover, we assume that no TCP flow has to wait or is blocked, which resembles an M/GI/ ∞ queuing discipline. Thus, the number of currently active flows F follows a Poisson distribution given in Equation (6) with mean $E[F]$. The generated signal traffic at the SDN controller is a superposition of all `Packet_In` messages created by the subflows of the set of active TCP flows. This means, the total rate η at which `Packet_In` messages are generated is the sum of the rates of each active TCP flow. In case all TCP flows follow the same characteristics and have the same signal rate $\eta_{Y_i} = \eta_Y$, $\forall i \in F$, $E[\eta]$ can be computed directly from the expected number of active TCP flows $E[F]$.

$$\begin{aligned}
P(F = k) &= \frac{(\lambda E[B])^k e^{-\lambda E[B]}}{k!} \\
E[F] &= \lambda E[B] \\
\eta &= \sum_F \eta_{Y_i} \\
\text{If } \eta_{Y_i} &= \eta_Y, \quad \forall i \in F : \\
\eta &= F \cdot \eta_Y \\
E[\eta] &= E[F] \cdot \eta_Y = \lambda E[B] \eta_Y \quad (6)
\end{aligned}$$

¹We overload the notation T_0 for sake of simplicity, instead of formally defining a deterministic random variable T with $T(t) = P(T \leq t) := \Theta(t - T_0)$, where Θ refers to the Heaviside step function.

The occupancy of the flow table at the SDN switch, i.e., the number of entries in the table T , has to be expressed as sum of a random number of indicator variables. They indicate for each of the F active flows whether it is in the on-phase, and thus, a rule is stored in the flow table. The distribution of the occupancy of the flow table in case of $F = k$ active flows, i.e., $P(T = m | F = k)$, can be expressed by means of the Poisson binomial distribution as presented in Equation (7), where F_m is the set of all subsets of m integers that can be selected from $k \geq m$ integers. This formula can again be simplified with a binomial distribution in case all TCP flows follow the same characteristics, which also gives an expectation for T .

$$\begin{aligned}
T &= \sum_F 1_{Y_{i,on}} \\
P(T = m | F = k) &= \begin{cases} 0, & k < m \\ \sum_{M \in F_m} \prod_{i \in M} \rho_{Y_i} \cdot \prod_{j \in \bar{M}} (1 - \rho_{Y_j}), & k \geq m \end{cases} \\
\text{If } \rho_{Y_i} &= \rho_Y, \quad \forall i \in k : \\
T &= F \cdot 1_{Y_{on}} \\
P(T = m | F = k) &= \binom{k}{m} \cdot \rho_Y^m \cdot (1 - \rho_Y)^{k-m} \\
E[T] &= E[F] \cdot \rho_Y = \lambda E[B] \rho_Y. \quad (7)
\end{aligned}$$

D. Application Mixes Model

For application mixes ω , we again assume independent Poisson arrivals of flows with rate λ and an exponentially distributed flow duration with mean $E[B]$. For each flow, the type of application i will be determined uniformly randomly according to the fixed flow probability ω_i , $0 \leq \omega_i \leq 1$, $\sum_i \omega_i = 1$. The results for the application mix η_ω , ρ_ω can then be computed by the weighted sum of the signaling rates η_i or table occupancies ρ_i of the single application model for each application i , respectively.

$$\begin{aligned}
\eta_\omega &= \sum_i \omega_i \cdot \eta_i \cdot \lambda \cdot E[B] \\
\rho_\omega &= \sum_i \omega_i \cdot \rho_i \cdot \lambda \cdot E[B] \quad (8)
\end{aligned}$$

In Section IV, the deduced characteristics for a single and multiple TCP flows will be evaluated in a realistic environment. In particular, the rate of `Packet_In` messages at the SDN controller and the occupancy of the flow table at the SDN switch will be analyzed for the presented scenarios.

IV. EVALUATION

We evaluate the impact of different TCP flow characteristics and SDN time-outs on the SDN controller traffic and the SDN flow table occupancy in a realistic scenario. Studying the work in [4], we find Table I, which provides inter-arrival times of TCP flow characteristics of four diverse mobile applications. It can be seen that the mean inter-arrival times of packets $E[A]$ can be as low as tens of milliseconds, e.g., in case of the music streaming service Aupeo, but can also extend to the order of some seconds, e.g., in case of browsing the social network Twitter. Also, the coefficient of variation c_A is application-specific and rather low in case of video chat application Skype. In contrast, very bursty arrivals with high

c_A were measured for the game app Angry Birds and Aupeo. Thus, we will focus the evaluation on the observed ranges of $E[A]$ and c_A . To demonstrate the correctness of our results, they are cross-validated by simulation in Section V.

A. Substitute Arrival Processes

To imitate different general arrival processes for packets of a TCP flow, we adopt a two-moment substitution as proposed in [25]. This means, to obtain a desired expectation $E[A]$ and coefficient of variation c_A of packet arrivals, the following substitute distribution functions are used:

$$\begin{aligned} \text{Case 1: } 0 < c_A \leq 1 \\ A(t) &= \begin{cases} 0, & 0 \leq t < t_1 \\ 1 - e^{-(t-t_1)/t_2}, & t_1 \leq t \end{cases} \\ &\text{where } t_1 = E[A](1 - c_A) \text{ and } t_2 = E[A]c_A. \\ \text{Case 2: } 1 < c_A \\ A(t) &= 1 - p \cdot e^{-t/t_1} - (1 - p) \cdot e^{-t/t_2} \\ &\text{where } t_{1,2} = E[A] \left(1 \pm \sqrt{\frac{c_A^2 - 1}{c_A^2 + 1}} \right)^{-1} \\ &\text{and } p = E[A]/2t_1, \quad pt_1 = (1 - p)t_2. \end{aligned} \quad (9)$$

The advantage of these substitute arrival processes is their mathematical tractability, thus, the moments of A_1 and A_2 can be calculated based on three parameters $E[A]$, c_A , and T_0 as presented in Equation (10). This allows to obtain the signaling rate at the SDN controller from Equation (4) and the flow table occupancy from Equation (5).

$$\begin{aligned} \text{Case 1: } 0 < c_A \leq 1 \\ E[A_1] &= t_2 + T_0 + \frac{t_1 - T_0}{1 - e^{-(T_0-t_1)/t_2}} \\ E[A_1^2] &= t_2^2 + \frac{(t_1 + t_2)^2 - (T_0 + t_2)^2 e^{-(T_0-t_1)/t_2}}{1 - e^{-(T_0-t_1)/t_2}} \\ E[A_2] &= T_0 + t_2 \\ E[A_2^2] &= (T_0 + t_2)^2 + t_2^2 \\ \text{Case 2: } 1 < c_A \\ E[A_1] &= \frac{p(t_1 - (T_0 + t_1)e^{-T_0/t_1})}{1 - pe^{-t_1/t_1} - (1-p)e^{-t_1/t_2}} \dots \\ &\quad \dots + \frac{(1-p)(t_2 - (T_0 + t_2)e^{-T_0/t_2})}{1 - pe^{-t_1/t_1} - (1-p)e^{-t_1/t_2}} \\ E[A_1^2] &= \frac{p(2t_1^2 - ((T_0 + t_1)^2 + t_1^2)e^{-T_0/t_1})}{1 - pe^{-t_1/t_1} - (1-p)e^{-t_1/t_2}} \dots \\ &\quad \dots + \frac{(1-p)(2t_2^2 - ((T_0 + t_2)^2 + t_2^2)e^{-T_0/t_2})}{1 - pe^{-t_1/t_1} - (1-p)e^{-t_1/t_2}} \\ E[A_2] &= T_0 + \frac{pt_1 e^{-T_0/t_1} + (1-p)t_2 e^{-T_0/t_2}}{pe^{-t_1/t_1} + (1-p)e^{-t_1/t_2}} \\ E[A_2^2] &= T_0^2 + \frac{2pt_1(T_0 + t_1)e^{-T_0/t_1}}{pe^{-t_1/t_1} + (1-p)e^{-t_1/t_2}} \dots \\ &\quad \dots + \frac{2(1-p)t_2(T_0 + t_2)e^{-T_0/t_2}}{pe^{-t_1/t_1} + (1-p)e^{-t_1/t_2}} \end{aligned} \quad (10)$$

B. Single Flow Model Analysis

Using the above described substitute arrival processes, we analyze the resulting SDN controller traffic and the SDN

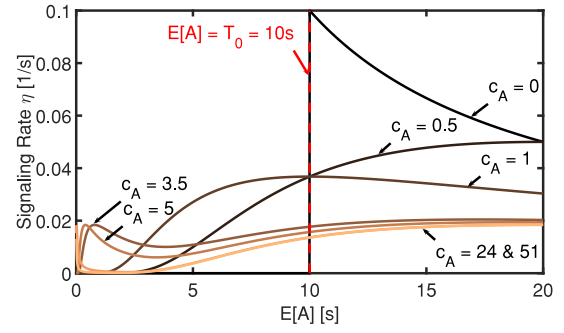


Fig. 5. Arrival rate η of Packet_In messages at SDN controller for fixed $T_0 = 10$ s depending on characteristics of TCP flow, i.e., the packet arrival process A .

flow table occupancy for different packet arrival streams and time-outs.

Figure 5 shows the arrival rate η of Packet_In messages originating from a single TCP flow on the y-axis. The mean packet inter-arrival time $E[A]$ is depicted on the x-axis, and the different curves depict the results for different coefficients of variation c_A ranging from 0 (black) to 51 (yellow). The time-out T_0 is set to 10s, which is a typical default value set by SDN controllers, e.g., NOX [31], and is also used as the default value in the Stanford OpenFlow deployment and DevoFlow [32].² In the deterministic case ($c_A = 0$), no flow ever times out when its packet inter-arrival time $E[A]$ is lower than T_0 , and thus $\eta = 0$. However, if $E[A] > T_0$, every packet will start a new subflow, but because $\eta = 1/E[A]$ in this case, the traffic at the SDN controller will decrease with increasing $E[A]$. For very large $E[A] \gg T_0$ (not depicted), all curves will eventually approximate $\eta = 1/E[A]$, as each arrival is increasingly more likely to be larger than T_0 and start a new subflow. For $0 < c_A < 1$ and $c_A = 1$, the signaling rate increases monotonically to a maximum before the rate merges ($0 < c_A < 1$, e.g., $c_A = 0.5$ at $E[A] = 20$) or converges ($c_A = 1$) towards $\eta = 1/E[A]$, respectively. In the hyperexponential case ($c_A > 1$), we observe that η has a first local maximum for small $E[A]$, then decreases to a local minimum, and increases to a second maximum before it eventually converges for large $E[A]$. The higher c_A , the more the first maximum is shifted to smaller $E[A]$, and the smaller the first and second maxima. With increasing c_A , the curves converge, which can be seen from the overlap of $c_A = 24$ and $c_A = 51$ for $E[A] > 3$. The envelope of these curves gives the maximum η for $E[A] < T_0$ independent of c_A . Thus, we see that different TCP flow characteristics influence the arrival rate of Packet_In messages at SDN controller in case of a fixed T_0 . In the interesting region for $E[A]$ smaller or slightly higher than T_0 , we observe that burstiness can decrease the SDN controller traffic for high $E[A]$, while burstiness increases η for flows with small $E[A]$.

Taking a look at the impact of the time-out T_0 , we fix $E[A] = 0.55$ s (see Skype in Table I) in Figure 6. The time-out T_0 is varied on the x-axis and the different curves indicate again different coefficients of variation c_A of the packet arrivals

²The latest SDN controllers OpenDaylight and ONOS use different values in their default configuration: 1800 seconds and ∞ , respectively.

TABLE I
MEAN INTER-ARRIVAL TIME $E[A]$ AND COEFFICIENT OF VARIATION c_A
OF PACKET ARRIVALS FOR DIFFERENT APPLICATIONS [4]

Application	$E[A]$	c_A
Twitter	8.91	4.95
Skype	0.55	3.55
Aupee	0.06	51.00
Angry Birds	0.66	24.09

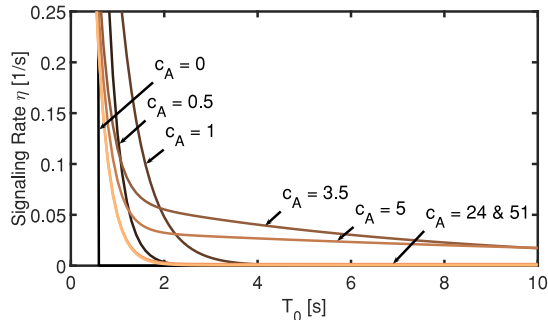


Fig. 6. SDN controller traffic arrival rate η depending on SDN switch time-out T_0 for fixed $E[A] = 0.55$ s.

in the TCP flow. All curves show a monotonically decreasing behavior towards 0 when T_0 becomes larger than $E[A]$. In the deterministic case $c_A = 0$, the signaling rate drops from a constant $\eta = 1/E[A] = 1.82$ to 0 at $T_0 = E[A]$, because no flow will time out if $T_0 > E[A]$. Starting from that asymptotic curve, for hypoexponential arrival processes ($c_A < 1$), the gradient will become smaller if c_A increases and the rates will converge slower towards 0. In the exponential and hyperexponential cases ($c_A \geq 1$), two intertwined effects cause the non-intuitive behavior visible in Figure 6 that the curves for very small and very high c_A show a fast convergence towards the asymptotic function, while the curves in between form an envelope and converge more slowly. First, when c_A increases from 1, the gradient of the curve transforms more quickly from a larger descent into a flatter slope. This will slow down the convergence towards 0, and can be seen when comparing the curves for $c_A = 1$, $c_A = 3.5$, and $c_A = 5$. At the same time, when c_A increases, the descent starts earlier, which brings the curves' points closer to the asymptotic function ($c_A = 0$). This will speed up again the convergence towards 0 for high c_A and can be observed when comparing the curves for $c_A = 5$ and $c_A = 24$. The envelope function of this group of curves constitutes an upper limit for the signaling traffic for given $E[A]$ and T_0 .

Based on these two figures several observations can be made concerning the dimensioning of T_0 . As long as $E[A] < T_0$ the signaling rate of an application flow is acceptably small, especially for $E[A] \ll T_0$. The coefficient of variation c_A only seems to play a minor role in these constraints, especially for really high values of c_A the signaling load is negligible. Therefore, controller interaction for processing this flow is kept to a minimum. In general, a higher c_A of an applications flow renders lesser load on a controller.

The flow table at the SDN switch is occupied by a flow rule, when the TCP flow is in an on-phase, i.e., a subflow of the TCP flow has not timed out. Figure 7 depicts the table occupancy

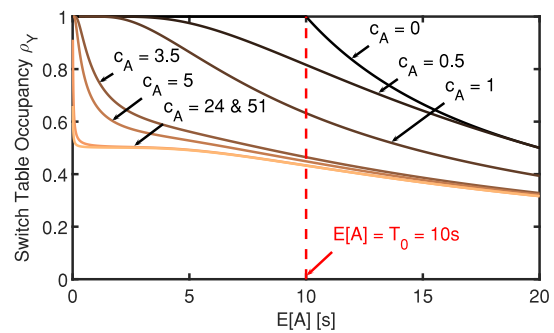


Fig. 7. SDN switch flow table occupancy ρ for time-out $T_0 = 10$ s and different TCP flow characteristics.

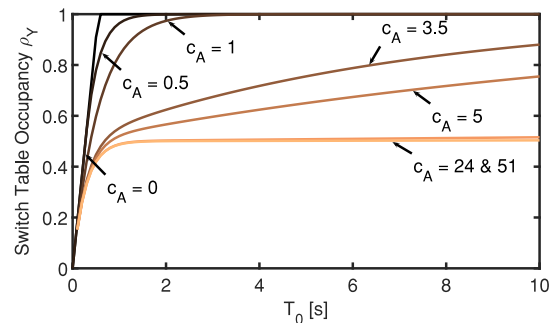


Fig. 8. Impact of time-out T_0 on SDN switch flow table occupancy of a single TCP flow with $E[A] = 0.55$ s.

ρ for an SDN switch with time-out $T_0 = 10$ s depending on the mean packet inter-arrival time $E[A]$, which is plotted on the x-axis, again for different values of c_A . Two effects can be clearly seen. First, all curves are monotonically decreasing, such that a higher $E[A]$ leads to a lower occupancy ρ in the flow table. This is due to the fact that a higher $E[A]$ increases the probability of a flow time-out when the next packets arrives later than T_0 . Larger $E[A]$ will also contribute to longer off-phases, which decreases the flow table occupancy. Second, the higher c_A , i.e., the more bursty the packet arrival process, the lower ρ , because of longer periods between two bursts, which will more likely cause a flow time-out and a long off-phase. In the extreme case of $c_A = 0$, the occupancy ρ is 1 if $E[A] < T_0$, and decreases hyperbolically with $\rho = \frac{T_0}{E[A]}$ if $E[A] \geq T_0$, which is the asymptotic function in this plot. In the hypoexponential case ($c_A < 1$), the larger the deterministic share of the substitute process (i.e., the smaller c_A), the sooner the convergence occurs. In the plot, the convergence for $c_A = 0.5$ is visible, when the curve overlaps with the asymptotic function for $E[A] > 18$. Eventually, the higher c_A , the earlier the drop of table occupancy and the more inert the convergence towards the asymptotic function.

Figure 8 investigates the impact of the time-out T_0 on the flow table occupancy of a single TCP flow for a fixed $E[A] = 0.55$ s. The x-axis shows the time-out T_0 , and the y-axis presents the resulting occupancy ρ for different c_A . The smaller T_0 , the more often TCP flows will time out and free the occupied space in the flow table. It can be seen that the choice of T_0 has more impact for TCP flows with small coefficient of variation. The resulting occupancies for small c_A range up to 1, i.e., there are time-outs T_0 , for which the flow

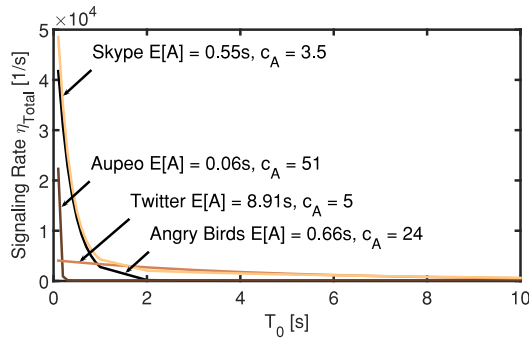


Fig. 9. Composite signaling rate at SDN controller depending on flow time-out T_0 for the four applications in the evaluation scenario.

rule will never be discarded. For TCP flows with high c_A , the occupancy will increase very slowly for increasing T_0 because the flows are generally more likely to time out. For example, a flow with $c_A = 51$ does not reach a higher occupancy than 50% throughout the investigated range of T_0 .

Figures 7 and 8 demonstrate that a change in the parameter T_0 also has a significant impact on the switch table occupancy ρ . The general conclusion is that a lower T_0 value decreases the occupancy. Keeping the previous results for the signaling rate η in mind, it is beneficial to choose a trade-off between signaling rate and table occupancy. For the values investigated, $T_0 = 3$ s offers a good solution: independent of the mean inter-arrival time $E[A]$ of an application and its coefficient of variation c_A , both signaling load and table occupancy are at acceptable levels. A higher T_0 would lead to a higher occupancy, a lower value to higher signaling towards the controller, which, in turn, may bring up another undesirable effect: overload at the controller.

C. Trade-Offs Between Signaling Load and Table Occupancy in the Case of Multiple Flows

At the SDN switch, typically multiple TCP flows arrive, which will contribute to the signaling rate at the SDN controller and the occupancy of the flow table. In this section, we will investigate this behavior and the resulting trade-offs for the four apps described above (see Table I). From a previous work [33], we have taken several numerical values for the composite model. Based on an extensive measurement of Internet access in dormitories, the authors observed an arrival rate of TCP flows of $\lambda = 158.73 \frac{1}{s}$ and a mean TCP flow duration of $E[B] = 234.95$ s. The numbers from [33] are used in our composite M/M/ ∞ model to compute the average number of active TCP flows $E[F] = 37293.6$ in the evaluation scenario. In the following figures, we will consider the simple case that all TCP flows are from the same type of application.

Figure 9 shows the composite signaling rate η_{Total} at the SDN controller in the evaluation scenario depending on the time-out T_0 . It can be seen that a very low time-out value T_0 will cause a significant amount of signaling at the controller, which will put it at risk of overload. Especially, applications like Aupeo, Angry Birds, and Skype will often time out and start a new subflow, which results in frequent `Packet_In` messages at the SDN controller. However, we see that, for the

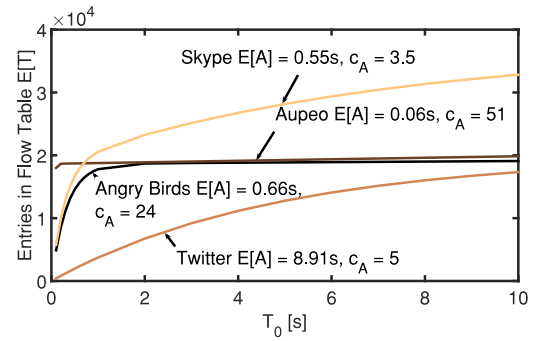


Fig. 10. Flow table occupancy at SDN switch depending on flow time-out T_0 for the four applications in the evaluation scenario.

TABLE II
SHARE-RATIO OF THE FOUR APPLICATIONS COMPARING THEIR GOOGLE PLAYSTORE DOWNLOADS

Application	Share-Ratio
Twitter	32%
Skype	35%
Aupeo	15%
Angry Birds	18%

bursty applications Aupeo and Angry Birds, a large enough $T_0 \geq 2$ s will make sure that the SDN controller traffic becomes very low. The signaling rate caused by applications like Twitter and Skype will decrease more slowly when T_0 increases. Nevertheless, a higher time-out value T_0 generally decreases the traffic at the SDN controller. Thus, the default value $T_0 = 10$ s is a good choice to relieve the SDN controller.

Figure 10 investigates the flow table occupancy in the given scenario for different T_0 . In general, the flow table occupancy increases with the time-out T_0 , as TCP flows are discarded later from the table. This results in the monotonic increase of all curves in the figure. Low table occupancies can only be achieved by very low T_0 . We see that the less bursty applications like Skype and Twitter continue to have a high gradient when T_0 increases. Thus, setting a high T_0 will cause a high table occupancy for these applications. In contrast, the occupancy of bursty applications Angry Birds and Aupeo only increases very flatly for high enough $T_0 \geq 2$ s.

All in all, we see that a very low T_0 is required to reach a low occupancy of the flow table. However, this will cause a huge signaling rate at the SDN controller. Vice versa, a high T_0 will cause a low signaling rate but a high table occupancy. Still, some room for trade-off is left by setting T_0 to a value around 2-3s. For the investigated applications Twitter and Skype, this will result in a reduction of the flow table occupancy compared to the default time-out of $T_0 = 10$ s, but will only cause a negligible increase of signaling at the SDN controller. Bursty applications like Aupeo and Angry Birds, are not negatively affected much by such choice of T_0 .

D. Application Mix Model

Usually not only one application is active within a network but a whole mix of applications. Therefore, a mix of the four introduced applications is evaluated. In order to create a fix application mix, the number of downloads in the beginning of

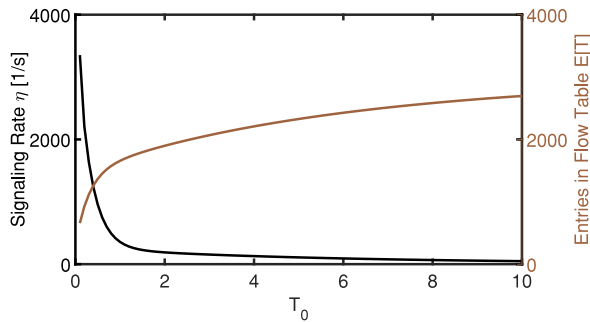


Fig. 11. Signaling rate & flow table occupancy for varying T_0 values in the case of an application mix.

December 2016 from Googles Playstore [34] for each application have been taken and put into relation. The resulting share-ratio amongst the applications is depicted in Table II. Twitter has a share-ratio of 32 %, Skype one of 35 %, Aupeo, though running out of service on December 16th 2016, a ratio of 15 %, and, last, Angry Birds a ratio of 18 %. In Figure 11 the signaling rate η_ω and the switch table occupancy ρ_ω of this previously mentioned application mix for a range of T_0 values is depicted. The x-axis shows T_0 ranging from 0 to 10 seconds, the y-axis on the left side the signaling rate from 0 to 4000 signals per second, and the y-axis on the right side the table occupancy from 0 to 4000 entries. As anticipated by the results for a single application, increasing the T_0 value has an enormous effect on the signaling rate towards the controller. With a T_0 equal to 0, each flow times out for each arriving packet, and, thus, the signaling rate is here at its highest value of around 3300 signals per second. Higher T_0 values lead to a lesser ratio of packet inter-arrival times from this application stream, and, thus, a lower frequency of flow deletions from the flow table. As the flow remains for a longer time inside the flow table, lesser signaling to the controller is required. Therefore, the signaling rate continuously decreases in an exponential manner to a value of 50 signals per second for $T_0 = 10$ s. For the switch table occupancy a coherent behavior is shown. With a small T_0 value below 1 second almost each new arriving packet of this application mix is arriving after the time-out has passed. Thus, the number of entries is low, e.g., only 650 for $T_0 = 0.1$ s. With a further increasing T_0 the flows remain for a longer time in the flow table. Therefore, the increase of the flow table occupancy flattens. Overall the table occupancy behavior has the resemblance of a logarithmic function. In the range of 0 to 1 seconds, the table occupancy increases heavily in this region from 600 to 1600. Beyond that the table occupancy only increases from 1600 to 2600 in the range of 1 to 10 seconds. In sum, for this application mix scenario, $T_0 = 3$ s is again the most suitable choice. With values below 3s the signaling rate is too high, with values beyond 3s the table occupancy is only increasing marginally.

E. Lessons Learned for Dimensioning T_0

Figure 12 summarizes the above findings by opposing signaling rate η and table occupancy ρ_Y for fixed $E[A] = 0.55$ s. Four qualitatively different cases are distinguished. Figure 12a

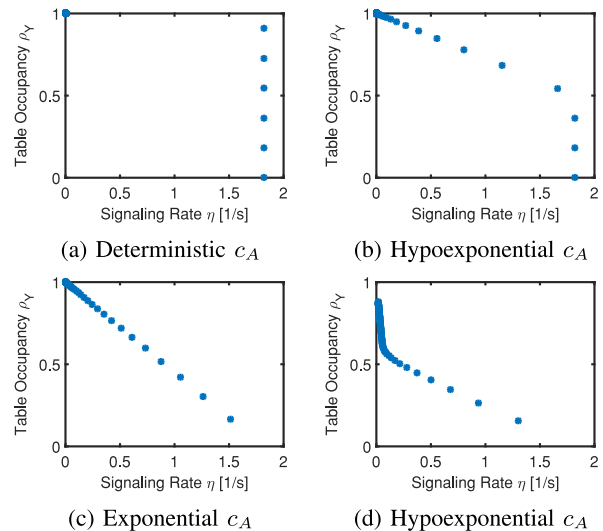


Fig. 12. Joint evaluation of signaling rate η and table occupancy ρ_Y for fixed $E[A] = 0.55$ s and $c_A = [0, 0.5, 1, 3.5]$.

shows the deterministic case with $c_A = 0$. If the time-out T_0 is larger than $E[A]$ the signaling rate is 0 and the table occupancy is 1. The smaller the time-out is set, the less the table occupancy. However, the signaling rate will not be affected by the choice of T_0 . For the hypoexponential example $c_A = 0.5$, Figure 12b illustrates that the choice of T_0 influences both η and ρ_Y . Starting from $\eta = 0$, $\rho = 1$, a decreasing T_0 will slowly decrease the table occupancy, but faster increase the signaling rate. When the signaling rate has reached its maximum, lowering T_0 will still decrease the table occupancy. While $c_A = 1$ in Figure 12c shows a balanced behavior, it constitutes the transition to the hyperexponential case for which the example $c_A = 3.5$ is presented in Figure 12d. Here, we see that decreasing T_0 can significantly reduce the table occupancy ρ_Y , while only negligibly affecting the signaling rate η . Only if T_0 becomes too small, the signaling rate will increase. As most applications produce traffic with hyperexponential c_A this behavior can be exploited.

All in all, a smaller T_0 decreases the switch table occupancy whilst increasing the signaling load on the controller. The biggest optimization potential can be observed especially for flows with small, hyperexponential c_A . If the time-out T_0 was optimized for these applications, the highest gain of table occupancy could be reached. Revisiting the results described above, a trade-off between these two metrics can be found for $T_0 = 2-3$ s: beyond that point, the switch table occupancy only increases marginally (in average), whilst the signaling load is at an acceptable minimum and decreases in small terms.

In general, an application specific T_0 value would be preferable, though a smaller T_0 value already offers a good starting-point. For setting an application specific value, additional traffic characterization mechanisms have to be deployed within the network. A possible integration could start with a small T_0 value whilst the application is still unknown and not enough packets were yet received. After successful characterization, the T_0 value can be changed dynamically. Another influencing factor on the current T_0 values should be the

overall table occupancy of a switch. If a flow table is full, no new flow can be installed. As most current SDN-controllers do not have a failure handling for this case, they simply retry to install that flow for each incoming packet until the action completes or no more packets of a flow arrive at the switch or at the controller.

A beneficial factor for the controller load could also be to enable caching at the switches. As soon as a flow times out due to its T_0 value, it could be marked as `to delete`, but yet still left active within the flow table. Now, if a packet matching that flow arrives again at the switch, the packet can immediately be forwarded and the entry and its timer can be reset to the initial setting. This would reduce the controller load, nevertheless, the controller should be notified, such that it maintains a coherent view of the network. If the flow table is full, `to delete` entries can be deleted from the flow table and replaced by new rules. Caching algorithms for flow tables in an SDN environment have been researched by various authors, as presented in Section II. Nevertheless, an analytic approach has not been taken yet.

V. MODEL VALIDATION

To validate the presented model, a discrete event simulation was implemented in MATLAB. Based on the results from [4], packet arrivals were simulated for each application. This means, random packet arrival times were obtained from the substitute arrival processes presented in Section IV-A for the given mean inter-arrival time $E[A]$ and coefficient of variation c_A . The packet arrivals of each flow allow to exactly determine the signaling and the flow table occupancy at the SDN controller depending on the time-out T_0 .

The simulation implements Poisson arrivals of flows with arrival rate $\lambda = 158.73 \frac{1}{s}$ as measured in [33]. However, using exponentially distributed flow durations with the mean presented in [33] ($E[B] = 234.95s$) proved to be much too short to reach the desired $E[A]$ and c_A given in [4] within the simulation. Therefore, for the validation of the model, the mean of the exponentially distributed flow duration is set to $E[B] = 3600s$ (one hour), and a minimum flow duration of 900s (15 min) is introduced. These modifications allowed the average $E[A]$ and c_A of the flows, which were created by the substitute packet arrival processes, to be close to the values described in [4].

In the validation scenario, the simulation duration is set to 50000s (13.89 hours). The presented results show the mean overall signaling rate and the flow table occupancy as observed by the SDN controller during five simulation runs. Additional to the mean results, also the 95% confidence intervals are shown. To account for the transient phase of the simulation, only the last 25000s of a simulation run have been evaluated. As all simulation runs converged to the stationary phase typically after around 10000s simulated time, the presented results can be assumed to provide a consistent view of the system in the long run. This long convergence time of the initially empty system occurred especially for simulation runs with a high coefficient of variation c_A . In these cases, the high burstiness of the packet arrivals and the resulting high

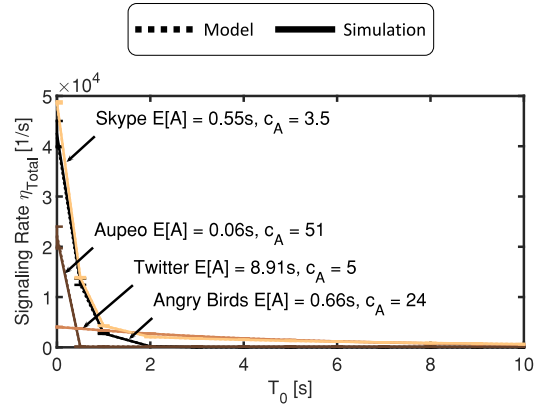


Fig. 13. Validation of multiple flow model for signaling rate.

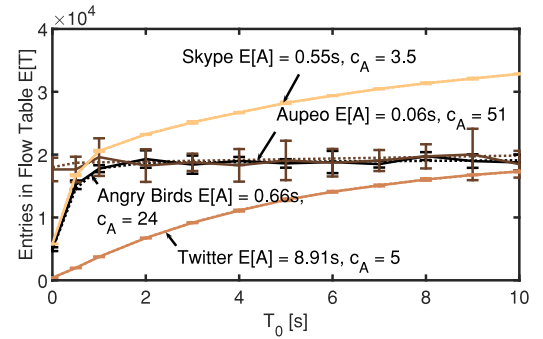


Fig. 14. Validation of multiple flow model for table occupancy.

variability of the flows' on- and off-phases slow down the convergence.

A. Single Flow Model

For the single flow model, we reproduce the results presented in Figures 5 to 8. This means, we simulate one flow each with different mean packet inter-arrival times ranging from 0 to 20s for a time-out $T_0 = 10s$. Additionally, we simulate one flow with $E[A] = 0.55$ (see Skype) and varying T_0 from 0 to 10s. All presented coefficients of variation were used (0, 0.5, 1, 3.5, 5, 24, 51). Note that for this validation, the simulated single flows span the whole simulation duration, and thus, the application characteristics presented in [4] could be exactly replicated. As the curves obtained from the simulation are perfectly aligned with the respective curves in Figures 5 to 8, we conclude that the single flow model is accurate and refrain from presenting the plots.

B. Multiple Flow Model

Figures 13 and 14 show the results of the validation of the multiple flow model, and correspond to Figures 9 and 10. In Figure 13 the results for both simulation (solid) and model (dotted) for the signaling rate is shown. Here, the simulation is matching the model perfectly, as both results for all applications align. Additionally, the confidence intervals are between very small for low T_0 values and not visible for higher T_0 values, confirming the validity of our results. In Figure 14 results for the table occupancy in the case of multiple flows

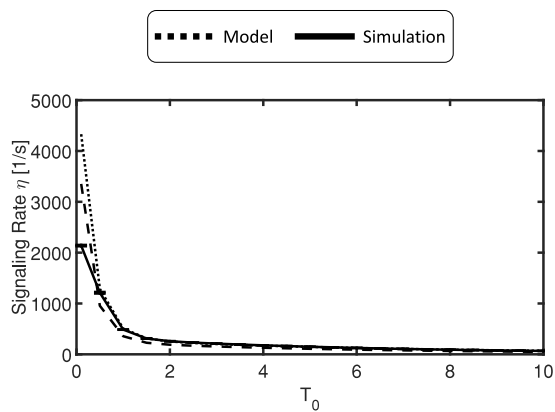


Fig. 15. The impact of the T_0 value on the signaling rate η for the case of a mixed application stream.

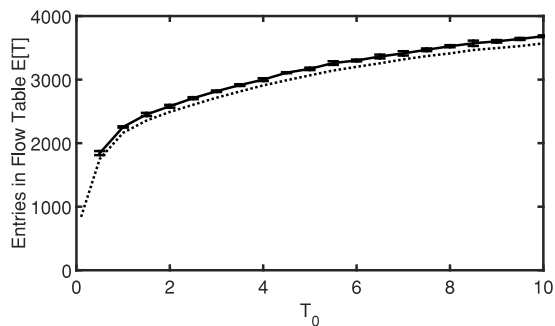


Fig. 16. The impact of the T_0 value on the flow table occupancy for the case of a mixed application stream.

for the introduced applications are presented. For the applications with a small c_A (Skype and Twitter) both simulation and model align perfectly. For Angry Birds and Aupeo with a very high c_A of 24 and 51 the limitations of the simulation are visible. The alignment between them is lower and the confidence intervals are bigger. This deviation in the results can be explained by the challenges presented at the beginning of this section.

C. Application Mix Model

To validate the model for a traffic mix of multiple applications, the application mix introduced in Section IV-C is simulated. In Figures 15 and 16, the dotted line shows the model results, and the solid line the mean results of the simulation and 95 % confidence intervals after five runs. The small confidence intervals indicate a consistent behavior for all simulation runs. The model provides accurate results for which the deviation never surpasses 10 % throughout the whole parameter range of T_0 . The remaining deviation can be attributed to the randomness within the single flows and the inadequacy to consistently reproduce the desired application characteristics for all flows, which propagate to the aggregated results. Still, the results confirm that the application mix model is sufficiently accurate to describe the signaling rate and table occupancy of an SDN controller.

In the following, it will be investigated how a variation of the application mix impacts the SDN controller. Instead

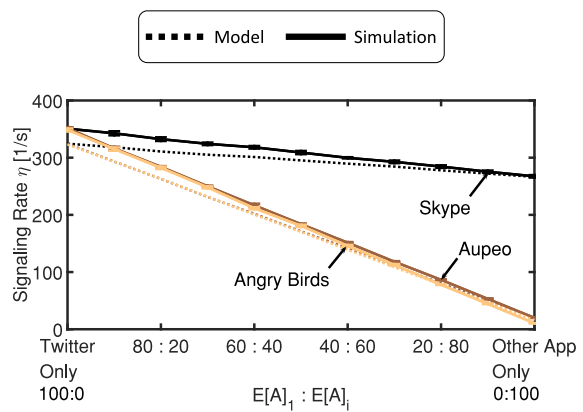


Fig. 17. Signaling rate η of multiple application mixes with varying percentage.

of assuming one application mix with a constant flow ratio between the applications, we now evaluate application mixes of two applications for varying ratios. To better compare the impact of a specific application, we fix Twitter in the application mix, and mix it with the remaining three applications. For these scenarios we use the time-out $T_0 = 3$ s, which we found earlier to be a good trade-off between signaling rate and table occupancy for application mixes in Section IV-D.

Figure 17 shows the resulting signaling rate at the SDN controller for the considered mixes of two applications. Thereby, the application mix is depicted on the x-axis, starting from 100 % Twitter flows and 0 % other app traffic, gradually decreasing Twitter until 0 % Twitter and 100 % other app traffic flows. The y-axis presents the signaling rate from 0 to 400 Packet_In messages per second. Here, again, the dotted line displays the model results and the solid line the mean results and 95 % confidence intervals of our simulation after 5 runs. The Twitter values always start at 320 signals per second for the modeling results, and 350 signals per second for the simulation, respectively. The discussed inadequacy of the simulation, which only reproduces the desired application characteristics on average, reappears here. Increasing the ratio of the other applications Skype, Angry Birds, and Aupeo, softens these effects, because these application characteristics can be more easily reached in the validation scenario. For example, increasing the ratio of Skype will decrease the signaling rate at the SDN controller. Both, simulation and adapted model, produce a signaling rate of 280, decreasing the deviation of the simulation down to almost 0 %. For Aupeo and Angry Birds the same linear course of the signaling rate can be seen. The very high coefficient of variation of these applications decrease the signaling rate close to 15. Thus, for the given time-out of $T_0 = 3$ s, a higher ratio of such traffic will reduce the load of the SDN controller.

The corresponding results for the total flow table occupancy is presented in Figure 18. The x-axis again depicts the application mix, the y-axis shows the overall flow table occupancy in the SDN switch. For Twitter-only traffic the model produces a result of 1300 entries in the flow table, which is slightly below the actual 1500 entries obtained by the simulation. Increasing the ratio of the other applications again decreases the deviation

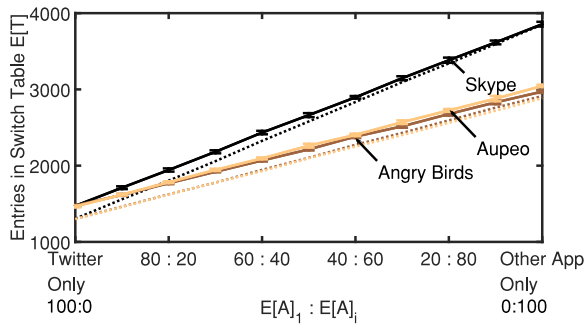


Fig. 18. Flow table occupancy ρ of multiple application mixes with varying percentage.

between model and simulation. Skype only traffic results in around 3700 entries on average, while Aupeo and Angry Birds only traffic will store almost 3000 entries in the flow table. As Twitter traffic uses the least flow table entries, again the trade-off between the signaling rate and the table occupancy is clearly visible.

To sum up, the simulative validation proved the accuracy and applicability of the single flow, the multiple flow, and the application mix models. Thereby, the small confidence intervals indicated a consistent behavior of the presented simulation results, which well align with the models. Thus, as they allow to produce accurate results for a given evaluation scenario much faster, the analytical models can be used to replace simulative approaches. Moreover, the impact of different application characteristics and the resulting trade-off between signaling rate and table occupancy could be confirmed. As the trade-off can be tuned by setting the time-out T_0 properly, our analytical models are a valuable tool for operators of SDN networks to find the optimal settings with respect to the specific application mixes in their networks.

VI. CONCLUSION

In this paper, an analytical model for SDN controller traffic and switch table occupancy is presented. The model focuses on the reactive operation mode of a controller. Incoming and unknown traffic at a switch therefore generates a request towards the controller. Based on this traffic, rules are created in the flow table of the switch, which specify the forwarding behavior. According to this flow table entry, further packets of this flow are processed by the switch only and do not require any further controller interaction, i.e., generate no signaling traffic. To avoid table overflows, unused entries are removed after a predefined time-out period T_0 . A time-out value is added to each flow entry. As soon as no packet has matched a flow for a duration of T_0 , the flow is automatically removed from the table. Any further packet of this flow triggers a new request towards the controller.

As both the switch table occupancy and the controller signaling can render a limiting factor to the forwarding speed of a network, a trade-off between signaling rate and switch table occupancy has to be found. The results in this paper deliver three main conclusions. First, with our presented model it is possible to calculate the effects of the discussed parameters

with respect to the performance of the SDN-based network. We start by modeling a single flow to understand its impact on the flow table occupancy and the resulting controller traffic. Based on these results, we adapt an $M/M/\infty$ queuing system and extend our model to understand the implications when multiple users, i.e., multiple flows, are in the system. Second, the presented discrete-event simulations validates the model in all scenarios, proving the accuracy and the applicability of our model. The impact of multiple application characteristics is additionally discussed. Third, it is shown that application specific parameters, such as the inter-arrival time of packets $E[A]$ and its coefficient of variation c_A , have a non-negligible impact on both the signaling rate and the table occupancy. Consequently, our results show that the default value of $T_0 = 10s$ is too large. This is comparable to the findings of Zarek *et al.* [16], which proposes a static time-out value of 5s. Based on our observations, the best trade-off could be reached by decreasing T_0 down to 2-3s. With these values, e.g., the flow table occupancy of Skype flows could be reduced by around 25%, while the controller traffic would only slightly increase. The best results, in terms of signaling rate and flow table occupancy, could be achieved for an application specific T_0 value, e.g., as presented by Vishnoi *et al.* [35]. This, however, renders the requirement for identifying applications or collecting flow statistics based on the packet stream, which can pose quite new challenges. Additionally, the model itself can also be applied to a composed network. However, the impact of switch-controller-interaction on the packet inter-arrival times of a flow are not covered by the current model. How this affects the accuracy in a composed network has to be covered by future research.

One might argue that the new generation of switches, has much bigger flow-table sizes, and, therefore, the importance of this work could decrease in the future. But, as the size of flow-tables increase, more fine-granular flow rules are possible, and, thus, the flow-table size could become an issue again.

REFERENCES

- [1] M. Jarschel, T. Zinner, T. Hossfeld, P. Tran-Gia, and W. Kellerer, "Interfaces, attributes, and use cases: A compass for SDN," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 210–217, Jun. 2014.
- [2] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turetli, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.
- [3] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
- [4] C. Schwartz, T. Hoßfeld, F. Lehrieder, and P. Tran-Gia, "Angry apps: The impact of network timer selection on power consumption, signalling load, and Web QoE," *J. Comput. Netw. Commun.*, vol. 2013, 2013, Art. no. 176217, doi: 10.1155/2013/176217.
- [5] C. Metter, M. Seufert, F. Wamser, T. Zinner, and P. Tran-Gia, "Analytic model for SDN controller traffic and switch table occupancy," in *Proc. 12th Int. Conf. Netw. Service Manag. (CNSM)*, Montreal, QC, Canada, 2016, pp. 109–117.
- [6] M. Jarschel *et al.*, "Modeling and performance evaluation of an OpenFlow architecture," in *Proc. 23rd Int. Teletraffic Congr.*, 2011, pp. 1–7.
- [7] K. Mahmood, A. Chilwan, O. N. Østerbø, and M. Jarschel, "On modeling controller-switch interaction in OpenFlow based SDNs," *Int. J. Comput. Netw. Commun.*, vol. 6, no. 6, p. 135, Nov. 2014.
- [8] K. Mahmood, A. Chilwan, O. N. Østerbø, and M. Jarschel, "Modelling of OpenFlow-based software-defined networks: The multiple node case," *IET Netw.*, vol. 4, no. 5, pp. 278–284, Sep. 2015.

- [9] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of OpenFlow-based software-defined networks based on queueing model," *Comput. Netw.*, vol. 102, pp. 172–185, Jun. 2016.
- [10] S. Azodolmolky *et al.*, "An analytical model for software defined networking: A network calculus-based approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2013, pp. 1397–1402.
- [11] A. G. Osgouei, A. K. Koohanestani, H. Saidi, and A. Fanian, "Analytical performance model of virtualized SDNs using network calculus," in *Proc. IEEE 23rd Iran. Conf. Elect. Eng.*, Tehran, Iran, 2015, pp. 770–774.
- [12] C. Lin, C. Wu, M. Huang, Z. Wen, and Q. Zheng, "Performance evaluation for SDN deployment: An approach based on stochastic network calculus," *China Commun.*, vol. 13, pp. 98–106, Sep. 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7560881/>
- [13] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 12, no. 5s, p. 77, 2016.
- [14] N. Beigi-Mohammadi, H. Khazaei, M. Shtern, C. Barna, and M. Litoiu, "On efficiency and scalability of software-defined infrastructure for adaptive applications," in *Proc. IEEE Int. Conf. Auton. Comput. (ICAC)*, Würzburg, Germany, 2016, pp. 25–34.
- [15] Y. Liu, Y. Li, Y. Wang, Y. Zhang, and J. Yuan, "On the resource trade-off of flow update in software-defined networks," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 88–93, Jun. 2016.
- [16] A. Zarek, Y. Ganjali, and D. Lie, "OpenFlow timeouts demystified," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2012.
- [17] T. Kim *et al.*, "A dynamic timeout control algorithm in software defined networks," *Int. J. Future Comput. Commun.*, vol. 3, no. 5, pp. 331–336, 2014.
- [18] H. Zhu, H. Fan, X. Luo, and Y. Jin, "Intelligent timeout master: Dynamic timeout for SDN-based data centers," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2015, pp. 734–737.
- [19] E.-D. Kim, S.-I. Lee, Y. Choi, M.-K. Shin, and H.-J. Kim, "A flow entry management scheme for reducing controller overhead," in *Proc. 16th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2014, pp. 754–757.
- [20] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "CacheFlow: Dependency-aware rule-caching for software-defined networks," in *Proc. Symp. SDN Res. (SOSR)*, Santa Clara, CA, USA, Mar. 2016, pp. 1–12, doi: 10.1145/2890955.2890969.
- [21] M. Kuźniar, P. Perešini, and D. Kostić, *What You Need to Know About SDN Flow Tables*. Cham, Switzerland: Springer, 2015, pp. 347–359. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15509-8_26
- [22] F. Ciucu and J. Schmitt, "Perspectives on network calculus: No free lunch, but still good value," in *Proc. ACM SIGCOMM Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, Helsinki, Finland, 2012, pp. 311–322.
- [23] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 42–55, 1998.
- [24] A. Feldmann, "Characteristics of TCP connection arrivals," in *Self-Similar Network Traffic and Performance Evaluation*. New York, NY, USA: Wiley, 2000, pp. 367–397.
- [25] H. Gold and P. Tran-Gia, "Performance analysis of a batch service queue arising out of manufacturing system modelling," *Queueing Syst.*, vol. 14, no. 3, pp. 413–426, 1993.
- [26] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence ten years of Internet traffic modeling," *IEEE Internet Comput.*, vol. 8, no. 5, pp. 57–64, Sep./Oct. 2004.
- [27] A. T. Andersen and B. F. Nielsen, "A Markovian approach for modeling packet traffic with long-range dependence," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 5, pp. 719–732, Jun. 1998.
- [28] N. G. Duffield and W. Whitt, "A source traffic model and its transient analysis for network control," *Stochastic Models*, vol. 14, nos. 1–2, pp. 51–78, 1998.
- [29] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Sel. Areas Commun.*, vol. 4, no. 6, pp. 833–846, Sep. 1986.
- [30] Specification-Version, ONF OpenFlow Switch, "1.4.0," Oct. 2013.
- [31] N. Gude *et al.*, "NOX: Towards an operating system for networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.
- [32] A. R. Curtis *et al.*, "DevoFlow: Scaling flow management for high-performance networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 254–265, 2011.
- [33] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet access traffic measurement and analysis," in *Proc. Traffic Monitor. Anal. (TMA) Workshop Co Located 13th Passive Active Meas. (PAM) Conf.*, Vienna, Austria, Mar. 2012.
- [34] Google. (2016). *Google Play*. Accessed on Dec. 5, 2016. [Online]. Available: <https://play.google.com/store>
- [35] A. Vishnoi, R. Poddar, V. Mann, and S. Bhattacharya, "Effective switch memory management in OpenFlow networks," in *Proc. 8th ACM Int. Conf. Distrib. Event Based Syst.*, 2014, pp. 177–188.



Christopher Metter received the master's degree in computer science from the University of Würzburg in 2014, where he is currently pursuing the Ph.D. degree. He is currently a Researcher with the Chair of Communication Networks, University of Würzburg. His research mainly focuses on reliability and availability of SDN.



Michael Seufert received the Diploma degree in computer science from the University of Würzburg in 2011, where he is currently pursuing the Ph.D. degree. He additionally passed the state examinations which are prerequisites for teaching mathematics and computer science in secondary schools. From 2012 to 2013, he has been with FTW Telecommunication Research Center Vienna where he has been working in the area of user-centered interaction and communication economics. He is currently a Researcher with the University of Würzburg. His research mainly focuses on QoE of Internet applications, QoE- and socially-aware traffic management solutions, monitoring and orchestration of edge cloud services, as well as performance analysis and modeling of communication systems.



Florian Wamser received the Diploma degree in computer science and the Ph.D. degree from the University of Würzburg, Germany, in 2009 and 2015, respectively, with the thesis on the topics characterization and modeling of application Internet traffic in broadband wireless access networks and the dissertation entitled *Performance Assessment of Resource Management Strategies for Cellular and Wireless Mesh Networks*. He is a Research Associate with the Chair of Communication Networks, University of Würzburg, Germany. He leads the group on cloud networks and Internet applications at the Chair of Prof. Dr.-Ing. P. Tran-Gia. His current research is focused on the analytical and simulative performance evaluation and optimization of cloud networks and related fields.



Thomas Zinner received the Diploma and Ph.D. degrees in computer science from the University of Würzburg, Germany, in 2007 and 2012, respectively. He is heading the research group on Next Generation Networks with the Chair of Communication Networks, University of Würzburg. His main research interests cover video streaming techniques, implementation of QoE awareness within networks, software defined networking and network function virtualization, and the performance assessment of these technologies and architectures.



Phuoc Tran-Gia is a Professor and the Director of the Chair of Communication Networks, University of Würzburg, Germany. He is also a member of the Advisory Board of Infosim (Germany) specialized in IP network management products and services. He has published over 100 research papers in major conferences and journals. He was a recipient of the Fred W. Ellersick Prize 2013 from the IEEE Communications Society.