Analytic model for SDN controller traffic and switch table occupancy

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

Angaben zur Veröffentlichung / Publication details:

Metter, Christopher, Michael Seufert, Florian Wamser, Thomas Zinner, and Phuoc Tran-Gia. 2016. "Analytic model for SDN controller traffic and switch table occupancy." In *12th International Conference on Network and Service Management (CNSM), 31 October - 4 November 2016, Montreal, QC, Canada*, edited by Noura Limam, Mohamed Cheriet, Mohamed Faten Zhani, Olivier Festor, Shannon Keith-Marsoun, and Carlos Raniery P. dos Santos, 109–17. Piscataway, NJ: IEEE. https://doi.org/10.1109/cnsm.2016.7818406.

Analytic Model for SDN Controller Traffic and Switch Table Occupancy

Christopher Metter, Michael Seufert, Florian Wamser, Thomas Zinner, Phuoc Tran-Gia
University of Würzburg, Germany
Email: {christopher.metter|seufert|wamser|zinner|trangia}@informatik.uni-wuerzburg.de

***Abstract*—Software Defined Networking (SDN) is a major paradigm in the field of current communication networks. SDN is used as the basis of many new networks although few performance models are available in the literature, and the majority of performance evaluations are based primarily on practical measurements. To fill this gap, we develop an analytical model to assess SDN control plane traffic as well as the occupancy of the flow table of an SDN switch. The contribution of this work is the formulation of the model for the performance-decisive parameters *control-plane traffic* and *flow table occupancy* and the application of the model for different data plane traffic characteristics. In the end, there is a discussion about the setting of time-out values for storing flow entries in the switch flow table depending on the traffic characteristics in the data plane. The trade-off between the signaling traffic in the control plane and the occupancy of the flow table is discussed to minimize both.**

I. Introduction

Software Defined Networking (SDN) is one of the major trends of current communication networks. It allows the decoupling of the control and data plane for the switching devices. With this simple principle, new networking scenarios are possible, facilitated by the introduction of increased control capabilities for access networks, data center networks, and metro networks [1, 2, 3].

The question is how traffic in SDN networks affects the SDN environment with controller, control plane, data plane, and switch. A performance analysis of the SDN technology is thus essential if the operator wants to understand the network in control and data plane, and yet, does not want to miss the improvements and advantages of the new technology.

From the perspective of performance analysis, analytical models are important to estimate and assess the performance of the control and data plane. In the end, they give guiding paradigms for managing, structuring, and deploying an SDN-based network. Closely linked to this, analytical models for SDN also enable a broader, more holistic understanding of the factors that influence the performance of SDN networks.

From a technical perspective, currently, two modes of operation are discussed for SDN-based 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. Consequently, 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 are 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 request to the controller. The controller defines a route through the topology and installs the new flow in the devices of the topology.

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 depending on the inter-arrival times of packets within the flow. If the packet inter-arrival time is larger than the time-out, the flow entry will be discarded. 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.

In this paper, an analytical model is given to analyze SDN in reactive mode. We investigate a scenario in which network flows are installed upon packet arrival by the controller and are discarded after a certain time-out period. We create an analytical model for 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. We then use these models and examine in detail the impact of different flow time-out values and their impact on the occupancy of a flow table in comparison to different data plane traffic types.

The contribution of this paper is threefold. We provide the analytical model as a tool ready to use for a network operator to analyze control plane traffic and flow tables utilization. Furthermore, we show the impact of traffic flow characteristics on control plane traffic. Finally, we extend the scenario to multiple concurrent flows in the network.

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. Finally, Section V concludes the paper.

II. Related Work

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

The authors of [4] 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.* extendend this work, as it is only viable for one single forwarding element and lacks correctness for highly bursty network traffic [5]. [6] created an OpenFlow-bases queuing model that provides the average packet sojourn time trough 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 have been conducted, matching the results. All three 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 [7]. 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 an 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 [8], 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.* point out the importance of suitable time-outs for each flow as well as a load awareness of the flow table in [9]. 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.* [10] choose an LRU caching algorithm to reduce the table-miss rate of the switch. Thus, the controller load can be reduced. In [11], 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.

[12] 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. In 1998 and 2000, Feldmann *et al.* presented fundamental work for modeling WAN traffic [13] [14]. 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 [15], 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 [16]. Andersen *et. al.* have created a model to represent these findings in superpositions of two-state Markov-modulated Poisson processes [17]. Whitt *et. al.* also present a candidate for source traffic models [18], [19].

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, cf. 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.

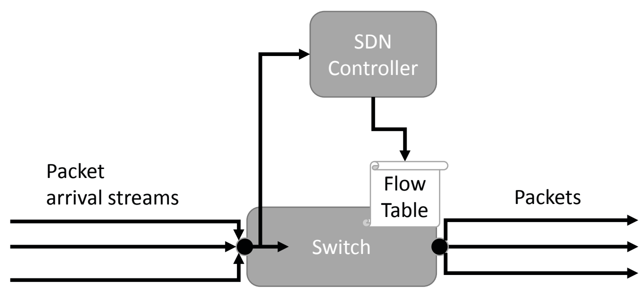


Fig. 1: SDN switching Model

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 by 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 0 indicates an infinite idle time-out (no idle time-out condition), any other value a time-out value in seconds [20].

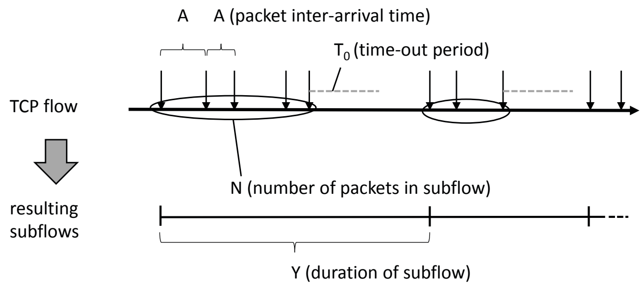


Fig. 2: Illustration of resulting subflows and their characteristics, i.e., number of packets N and duration Y, based on the inter-arrival time distribution A of the TCP flow and the time-out period T_0 of the SDN switch.

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 a new subflow.

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.

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

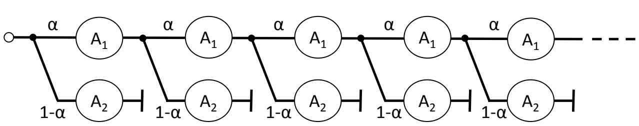


Fig. 3: Phase diagram for the composition of Y.

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:

$$Y = A_{1(1)} + A_{1(2)} + \ldots + A_{1(N')} + A_2 \tag{2}$$

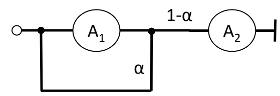


Fig. 4: Feedback loop for subflow duration Y.

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.

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

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.

$$\begin{aligned}
E[Y] &= -\Phi'_Y(0) = \frac{\alpha}{1-\alpha} E[A_1] + E[A_2] \\
\eta_Y &= \frac{1}{E[Y]} = \frac{1}{\frac{\alpha}{1-\alpha} E[A_1] + E[A_2]} \\
E[Y^2] &= \Phi''_Y(0) \\
&= \frac{2\alpha^2 E[A_1]^2}{(1-\alpha)^2} + \frac{2\alpha E[A_1]E[A_2] + \alpha E[A_1^2]}{1-\alpha} + E[A_2^2] \\
Var[Y] &= E[Y^2] - E[Y]^2 = \frac{\alpha E[A_1^2] - \alpha^2 Var[A_1]}{(1-\alpha)^2} + Var[A_2] \\
c_Y &= \frac{\sqrt{Var[Y]}}{E[Y]} \\
&= \frac{\sqrt{Var[A_2](1-\alpha)^2 + \alpha E[A_1^2] - \alpha^2 Var[A_1]}}{\alpha E[A_1] + (1-\alpha) E[A_2]}
\end{aligned} \tag{4}$$

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. 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 utilization 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.

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

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 $E[F]$.

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

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 flow 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.

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.

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

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 [21], we find Table I, which gives inter-arrival time 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 result, each result has been cross-validated by simulation. These results perfectly fit the analysis but have been omitted from the figures as the results only showed minor, insignificant differences.

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 [15]. 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:

$$\begin{aligned} &\text{Case 1: } 0 < c_A \leq 1 \\ &\qquad A(t) = \begin{cases} 0, & 0 \leq t < t_1 \\ 1 - e^{-(t-t_1)/t_2}, & t_1 \leq t \end{cases} \\ &\qquad \text{where } t_1 = E[A](1 - c_A) \text{ and } t_2 = E[A]c_A. \\ &\text{Case 2: } 1 < c_A \\ &\qquad A(t) = 1 - p \cdot e^{-t/t_1} - (1 - p) \cdot e^{-t/t_2} \\ &\qquad \text{where } t_{1,2} = E[A]\left(1 \pm \sqrt{\frac{c_A^2 - 1}{c_A^2 + 1}}\right)^{-1} \\ &\qquad \text{and } p = E[A]/2t_1,\ pt_1 = (1 - p)t_2. \end{aligned} \tag{8}$$

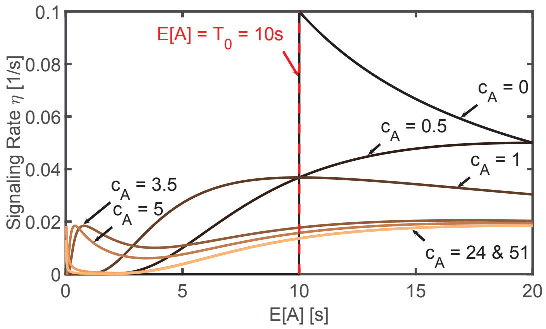


Fig. 5: Arrival rate η of `Packet_In` messages at SDN controller for fixed $T_0 = 10$s depending on characteristics of TCP flow, i.e., the packet arrival processs A.

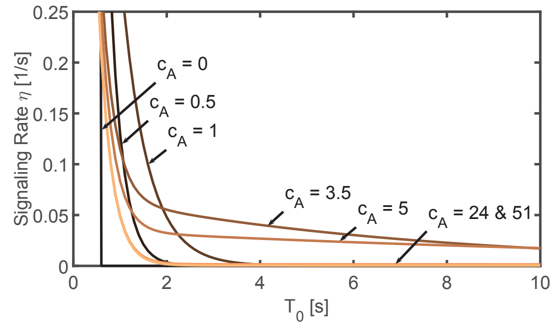


Fig. 6: SDN controller traffic arrival rate η depending on SDN switch time-out T_0 for fixed $E[A] = 0.55$s.

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 9. This allows to obtain the signaling rate at the SDN controller from Equation 4 and the flow table occupancy from Equation 5.

Case 1: $0 < c_A \leq 1$

$$E[A_1] = t_2 + T_0 + \frac{t_1 - T_0}{1 - e^{-(T_0 - t_1)/t_2}}$$
$$E[A_1^2] = t_2^2 + \frac{(t_1+t_2)^2 - (T_0+t_2)^2 e^{-(T_0-t_1)/t_2}}{1 - e^{-(T_0-t_1)/t_2}}$$
$$E[A_2] = T_0 + t_2$$
$$E[A_2^2] = (T_0+t_2)^2 + t_2^2$$

Case 2: $1 < c_A$

$$E[A_1] = \frac{p(t_1 - (T_0+t_1)e^{-T_0/t_1})}{1 - pe^{-t/t_1} - (1-p)e^{-t/t_2}} \cdots$$
$$\cdots + \frac{(1-p)(t_2 - (T_0+t_2)e^{-T_0/t_2})}{1 - pe^{-t/t_1} - (1-p)e^{-t/t_2}} \quad (9)$$
$$E[A_1^2] = \frac{p(2t_1^2 - ((T_0+t_1)^2 + t_1^2)e^{-T_0/t_1})}{1 - pe^{-t/t_1} - (1-p)e^{-t/t_2}} \cdots$$
$$\cdots + \frac{(1-p)(2t_2^2 - ((T_0+t_2)^2 + t_2^2)e^{-T_0/t_2})}{1 - pe^{-t/t_1} - (1-p)e^{-t/t_2}}$$
$$E[A_2] = T_0 + \frac{pt_1 e^{-T_0/t1} + (1-p)t_2 e^{-T_0/t2}}{pe^{-t/t_1} + (1-p)e^{-t/t_2}}$$
$$E[A_2^2] = T_0^2 + \frac{2pt_1(T_0+t_1)e^{-T_0/t1}}{pe^{-t/t_1} + (1-p)e^{-t/t_2}} \cdots$$
$$\cdots + \frac{2(1-p)t_2(T_0+t_2)e^{-T_0/t2}}{pe^{-t/t_1} + (1-p)e^{-t/t_2}}$$

B. Single Flow Model Analysis

Using the above described substitute arrival processes, we analyze the resulting SDN controller traffic and the SDN 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 [22], and is also used as the default value in the Stanford OpenFlow deployment and DevoFlow [23][1]. 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 $E[A]$.

[1]The 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 [21].

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

Taking a look at the impact of the time-out T_0, we fix $E[A] = 0.55$s (cf. 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 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 coverge 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 coverge 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.

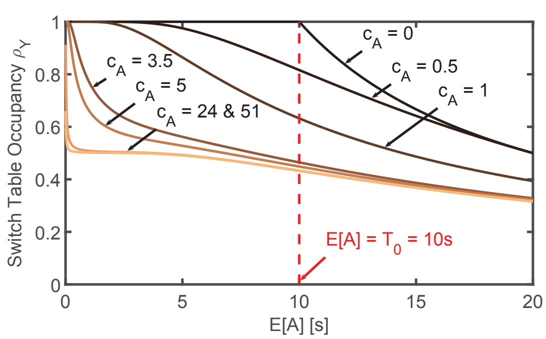


Fig. 7: SDN switch flow table occupancy ρ for time-out $T_0 =$ 10s and different TCP flow characteristics.

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 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 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, $T_0 = 3s$ offers a good solution: independent of the mean

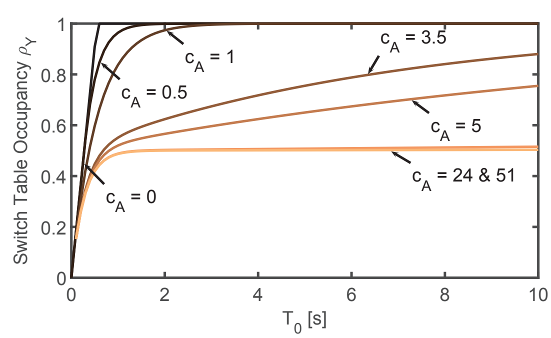


Fig. 8: Impact of time-out T_0 on SDN switch flow table occupancy of a single TCP flow with $E[A] = 0.55$s.

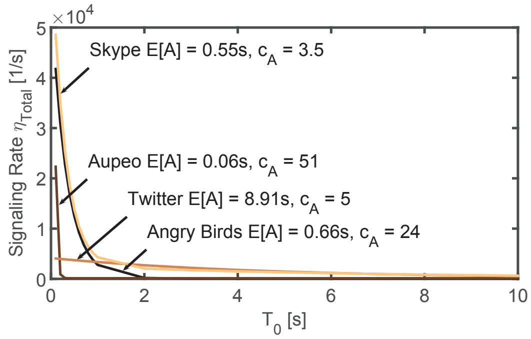


Fig. 9: Composite signaling rate at SDN controller depending on flow time-out T_0 for the four applications in the evaluation scenario.

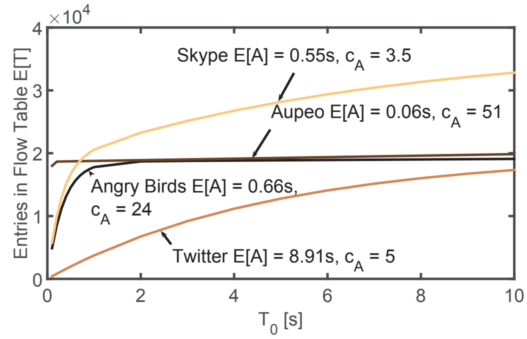


Fig. 10: Flow table occupancy at SDN switch depending on flow time-out T_0 for the four applications in the evaluation scenario.

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

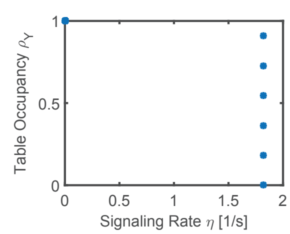
At the SDN switch, typically multiple TCP flows arrive, which will contribute to the signaling rate at the SDN controller and the occupancy of the flow table. In this section, we will investigate this behavior and the resulting trade-offs for the four apps described above (cf. Table I). From a previous work [24], 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 $\lambda = 158.73$ and a mean TCP flow duration of $E[B] = 234.95$s. The numbers from [24] 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 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.

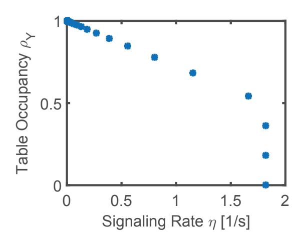
Figure 10 investigates the flow table occupancy in the given scenario for different T_0. In general, the flow table occupancy increases with the time-out T_0, as TCP flows are discarded later from the table. This results in the monotonic increase of all curves in the figure. Low table occupancies can 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.

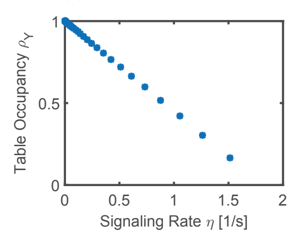
D. Lessons Learned for Dimensioning T_0



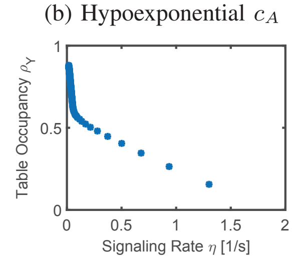
(a) Deterministic c_A



(b) Hypoexponential c_A



(c) Exponential c_A



(d) Hypoexponential c_A

Fig. 11: Joint evaluation of signaling rate η and table occupancy ρ_Y for fixed $E[A] = 0.55$s and $c_A = [0, 0.5, 1, 3.5]$.

Figure 11 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 11a 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 11b 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 11c shows a balanced behavior, it constitutes the transition to the hyperexponential case for which the example $c_A = 3.5$ is presented in Figure 11d. 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 $T_0 = 2 - 3s$: 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 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. Conclusion

This paper presented an analytic model of SDN controller traffic and switch table occupancy. The model focuses on the reactive operation mode of a controller, thus, incoming and unknown traffic at a switch generates a request towards the controller. With our presented model it is possible, for a given application with the packet arrival process parameters $E[A]$ and c_A, to analytically model these effects for both single and multiple flows. 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 adapted an $M/M/\infty$ queuing system and extended our model to understand the implications when multiple users, i.e. multiple flows, in the system.

The results, which have been cross-confirmed with simulations, deliver two main conclusions. First, 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. However, the time-out value T_0 introduces an opportunity for trade-off. Consequently, our results show that the default value of $T_0 = 10$s is too large. This is comparable to the findings of Zarek *et al.* in [7], 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.* [25]. 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 migth 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 and 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," *Communications Magazine, IEEE*, vol. 52, no. 6, pp. 210–217, June 2014.

[2] B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 3, pp. 1617–1634, 2014.

[3] D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[4] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and Performance Evaluation of an Openflow Architecture," in *Proceedings of the 23rd International Teletraffic Congress*. International Teletraffic Congress, 2011.

[5] K. Mahmood, A. Chilwan, O. N. Østerbø, and M. Jarschel, "On the modeling of openflow-based sdns: The single node case," *arXiv preprint arXiv:1411.4733*, 2014.

[6] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of openflow-based software-defined networks based on queueing model," *Computer Networks*, vol. 102, pp. 172–185, 2016.

[7] A. Zarek, Y. Ganjali, and D. Lie, "Openflow timeouts demystified," *Univ. of Toronto, Toronto, Ontario, Canada*, 2012.

[8] T. Kim, K. Lee, J. Lee, S. Park, Y.-H. Kim, and B. Lee, "A dynamic timeout control algorithm in software defined networks," *International Journal of Future Computer and Communication*, vol. 3, no. 5, p. 331, 2014.

[9] H. Zhu, H. Fan, X. Luo, and Y. Jin, "Intelligent timeout master: Dynamic timeout for sdn-based data centers," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 734–737.

[10] 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 *Advanced Communication Technology (ICACT), 2014 16th International Conference on*. IEEE, 2014, pp. 754–757.

[11] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Rule-caching algorithms for software-defined networks," Citeseer, Tech. Rep., 2014.

[12] M. Kuźniar, P. Perešíni, and D. Kostić, *What You Need to Know About SDN Flow Tables*. Cham: Springer International Publishing, 2015, pp. 347–359. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15509-8_26

[13] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet wan traffic," in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4. ACM, 1998, pp. 42–55.

[14] A. Feldmann, "Characteristics of tcp connection arrivals," *Self-similar network traffic and performance evaluation*, pp. 367–397, 2000.

[15] H. Gold and P. Tran-Gia, "Performance analysis of a batch service queue arising out of manufacturing system modelling," *Queueing Systems*, vol. 14, 1993.

[16] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence ten years of internet traffic modeling," *Internet Computing, IEEE*, vol. 8, no. 5, pp. 57–64, 2004.

[17] A. T. Andersen and B. F. Nielsen, "A markovian approach for modeling packet traffic with long-range dependence," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 5, pp. 719–732, 1998.

[18] N. Duffield and W. Whitt, "A source traffic model and its transient analysis for network control," *Stochastic Models*, vol. 14, no. 1-2, pp. 51–78, 1998.

[19] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *Selected Areas in Communications, IEEE Journal on*, vol. 4, no. 6, pp. 833–846, 1986.

[20] O. S. Specification-Version, "1.4.0," 2013.

[21] 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," *Journal of Computer Networks and Communications*, 2013.

[22] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, 2008.

[23] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: scaling flow management for high-performance networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 254–265.

[24] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet Access Traffic Measurement and Analysis," in *Traffic Monitoring and Analysis (TMA) workshop co-located with the 13th Passive and Active Measurement (PAM) conference*, Vienna, Austria, Mar. 2012.

[25] A. Vishnoi, R. Poddar, V. Mann, and S. Bhattacharya, "Effective switch memory management in openflow networks," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. ACM, 2014, pp. 177–188.