How fast can you reconfigure your partially deployed SDN network?

Christian Sieber, Raphael Durner, Wolfgang Kellerer
Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich, Germany
Email: {c.sieber,r.durner,wolfgang.kellerer}@tum.de

***Abstract*—Software-Defined Networking (SDN) enables fast reconfiguration of the network based on the decisions of a logically centralized controller. While SDN is in transition from laboratories and greenfield data-centers to enterprise networks, the need for managing partial SDN deployments arises. This incremental deployment of SDN has its own set of challenges, for example the design of the mixed control plane or the integration in existing network management systems. In this paper we focus on the different speed of reconfiguration between legacy and SDN devices and how those differences constrain the overall reconfigurability of the network. At first we present measurement results highlighting the differences between SDN and legacy devices in terms of reconfiguration speed. The measurements show an up to hundredfold difference of reconfiguration times between SDN and legacy devices. Second, we introduce a methodology which allows to evaluate the impact of these differences on the maximum reconfiguration rate of the whole network for different SDN deployment stages. Third, we investigate a large number of real world topologies and the gain in terms of reconfigurability that can be achieved by deploying SDN incrementally. The results show that even a small number of slow devices can severely constrain the maximum reconfiguration rate. Furthermore, even with the majority of the network nodes being replaced by SDN devices, the maximum reconfiguration rate increases in the best case only up to 5 times compared to the all-legacy network. Based on the findings we conclude that research has to focus more on the management of partial SDN deployments w.r.t. potentially large differences in reconfiguration times of the devices. Otherwise the potential of SDN lies dormant until the whole network is migrated to SDN-enabled devices.**

I. Introduction

Network switches are commonly feature-packed, monolithic and highly integrated devices with proprietary management interfaces. Routing decisions are made by statically configuring routes or by distributed routing protocols such as OSPF which are built into the devices. The administration and configuration of the algorithms and the devices in general is performed through a command line interface (CLI) via telnet or SSH or using NETCONF. To enable real-time packet processing in the switches, the static or dynamic routing decisions are translated by the switch operating system to simple forwarding rules and pushed to an application specific integrated network processor.

Software-defined networking (SDN) introduces a split between the packet forwarding, the *data plane*, and packet routing decisions, the *control plane* of network devices. Control decisions can be made from a logically centralized location, the *SDN controller*, and pushed directly to the network processor of the devices. OpenFlow is a protocol which implements the idea of SDN. Through OpenFlow, forwarding rules can be pushed to OpenFlow-enabled devices, ideally independent of the manufacturer of the device. This reduces complexity of network devices and promises to improve networking in multiple dimensions, from cost per device to near real-time insight into the network [1]. However, an operator is unlikely to replace all devices in his network with SDN-enabled devices at once and therefore there will be a transition period with the need to manage networks with partial SDN deployment.

Network management and configuration is mostly proactive and static in current legacy deployments. A new network is setup, configured and tested and afterwards the configuration stays unchanged for months. In contrast to that, SDN and OpenFlow provide new possibilities. The logically centralized view and a fast binary protocol facilitates reactive network management ideas where the configuration is changed upon specific events such as link failure, a sudden rise in traffic volume or even the arrival of specific network packets. Therefore, we expect the number of reconfigurations to considerably increase in modern SDN-enabled networks. For partial deployments we expect this to be true as well, as a network operator will want to profit from the investment in SDN devices even if not all devices in the network are replaced. An abstraction layer [2], [3] provides the necessary generalization of the network elements and a unified configuration interface.

One way to jointly manage legacy and SDN devices is to exploit legacy switch features for this purpose, e.g. BGP [4] or Access Control Lists [5]. In [6], the authors propose Panopticon which uses a special VLAN tagging schema on the legacy devices. VLAN tagging is also a feature which is available on all managed switches and therefore a suitable candidate for heterogeneous networks. In the remainder of this paper we assume a transition strategy based on VLAN tagging on the legacy devices.

Figure 1 depicts the abstraction layer. The figure shows a network consisting of legacy and SDN-enabled devices. This network is controlled by a unified control and management layer, referred to as Network Services Abstraction Layer (NSAL), which provides applications northbound of the NSAL, e.g. an orchestrator, control over the network. For

ISBN 978-3-901882-94-4 © 2017 IFIP

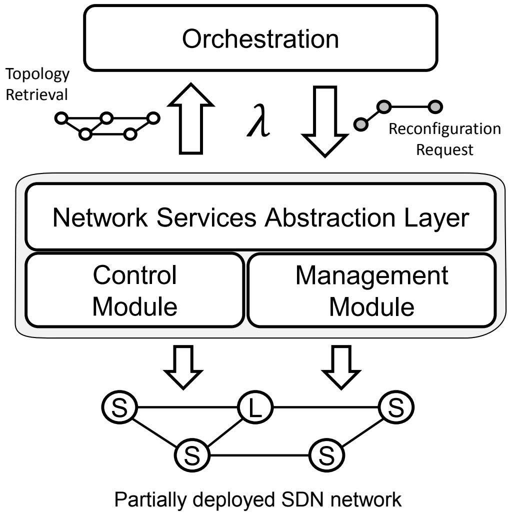

Figure 1: Overview of the investigated scenario. A network services abstraction layer (NSAL) [2], [3] provides the necessary generalization for combined legacy and SDN networks. Reconfigurations are done by network application or orchestration on top of NSAL with a rate of λ.

example, the NSAL could be implemented by OpenDaylight [7], a popular SDN controller with support for legacy devices, and the orchestrator could be OpenStack [8], a cloud computing framework. The orchestrator uses the abstractions provided by the NSAL to query the network topology and to trigger reconfigurations. We assume that these reconfigurations are requested with a certain rate λ by the orchestrator and implemented by the NSAL in the network on one or multiple devices. A problem arises if the global rate of reconfiguration is over a certain threshold the network can handle, denoted as $\lambda_{g,max}$. Hence, if $\lambda > \lambda_{g,max}$ the system becomes overloaded and reconfiguration times increase.

In this paper we focus on one particular aspect of a partial SDN deployment: The different reconfiguration timing characteristics of legacy and SDN devices. Motivated by the work in [9] where the authors show the impact of different timings of SDN-enabled devices on the data traffic in large-scale infrastructures, we investigate the impact of slow legacy devices on the overall reconfiguration capabilities of the network. The goal of this paper is to first, better understand how legacy and SDN-enabled devices differ in terms of reconfiguration time, second, develop a methodology for describing the reconfigurability of partial SDN deployments, and third, quantify how the reconfigurability is impacted by different ratios of SDN deployment.

The contribution of this work is threefold: First, we present measurements quantifying the differences in reconfiguration time between legacy and SDN devices. Second, we use an analytical approach based on queuing theory to quantify and compare the maximum reconfiguration rate λ_{max} based on the ratio of SDN and legacy devices in the network. Furthermore we propose an intuitive metric to compare different network topologies in terms of their suitability for SDN deployment. Third, we investigate a large number of real world topologies and the gain in terms of reconfigurability that can be achieved by increasing SDN deployment.

The remainder of this paper is structured as follows: In Chapter II we discuss our measurements of reconfiguration times of different network devices. In Chapter III we introduce the notations of our network model and our analytical approach. In Chapter IV we investigate a large number of real-world topologies based on the introduced methodology. In Chapter V we summarize and interpret the findings. In Chapter VI we give an overview over related work in this area of research and in Chapter VII we conclude this paper and give future research directions.

II. Measurements

The goal of the measurements is to gain insights into the reconfiguration times of legacy and SDN-enabled devices. We measure different SDN and legacy devices from multiple vendors, different years of release and sizes in our testbed. For legacy devices we include HP V1910 (for small organizations, 2010) and Cisco Catalyst 4500 (for campus access & distribution, 2007). For SDN-enabled devices we include two OpenFlow hardware switches Pica8-P3290 and Pica8-P3297 (for small cloud data centers, 2012). Additionally we include the NEC PF5240 (for data center, 2011) in the evaluation, which can be used in legacy and SDN-enabled mode, and the software switch Open vSwitch (OVS) in SDN-enabled mode.

Figure 2 shows our measurement setup. We use a tap device connected to the data plane and to the OpenFlow/management port. The tap is connected to a high precision measurement card with nanosecond precision. We define the reconfiguration time based on the data plane effect delay, i.e. from sending the reconfiguration command until the effect is visible on the data plane. For this we decode the incoming packets on the OpenFlow/management port and record the time of the last packet of the incoming reconfiguration. On the data plane, we generate a stream of marked UDP packets with constant inter-arrival time with a data-rate close to the interface's line rate. That way we can identify the exact packet when the reconfiguration was applied and compare it to the time of the reconfiguration command entering the device.

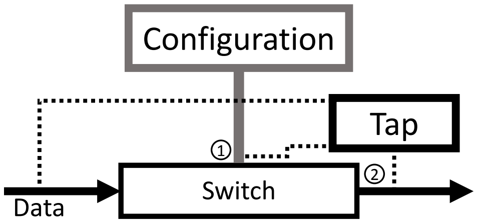

Figure 2: Measurement set-up consisting of a high-precision tap device, the self-developed reconfiguration agent and the switch to be measured. Timing characteristics are measured by tapping the OpenFlow/management port (1) and the data plane interfaces (2).

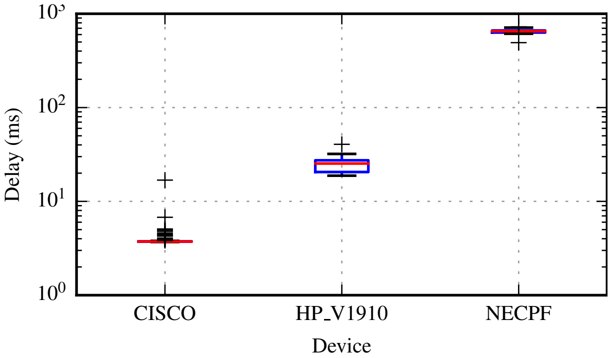

Figure 3: Reconfiguration times for VLAN tagging via telnet for the three legacy devices Cisco Catalyst 4500 (left), HP V1910 (middle) and NEC PF5240 (right).

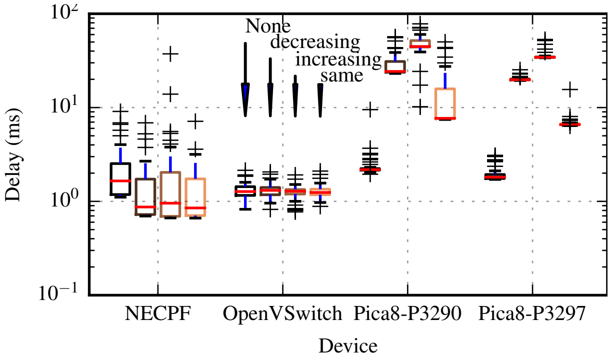

Figure 4: Rule push time for the four OpenFlow-enabled devices NEC PF5240, Pica8 P3290 & P3297 and OpenVSwitch grouped by table population strategy.

As measuring all possible combinations and parameters of all devices is infeasible, we focus for the measurements on the SDN transition strategy similar to Panopticon [6], which uses VLAN tagging to control legacy devices in a partial SDN deployment. For legacy network this is usually a text command which is sent to the switch. With the OpenFlow protocol, VLAN tagging can be set-up using OpenFlow flow modification messages. We use a custom OpenFlow Controller based on the libfluid [10] framework to populate the tables of the switches and install the rules that are measured. For the legacy devices, we use a custom configuration agent which accesses the devices via telnet. Each measurement is repeated between 50 and 100 times.

Figure 3 shows the measurement results for the evaluated *legacy* switches using the command to add VLAN tagging to a port. The data plane effect delay is shown as a box-plot with the median, 25 % and 75 % quartile and outliers. The results differ in two orders of magnitude from a median of 3.73 ms for the Cisco device to a median of 652 ms for the NEC. The HP device shows a median delay of 25.3 ms. The variance of the results is small for all three devices.

For measuring SDN devices there are more conditions to take into account. As other works have shown [11], the behavior of OpenFlow switches depends on the number of installed rules and the priority of the installed rules. Especially the priorities of the rules are important as higher priority rules mask lower priority rules and the switch has to make sure that the effects are independent of the order in which the rules are added. Therefore for some cases it can be necessary to search all current rules before the new one can be added. We measure the delay for four cases of flow table population:

1) **None**: empty flow table
2) **Decreasing**: 1000 installed rules with decreasing priority, the measured rule has the lowest priority.
3) **Increasing**: 1000 installed rules with increasing priority, the measured rule has the highest priority.
4) **Same**: 1000 installed rules with the same priority, the measured rule has also the same as the others

Figure 4 shows the measured delays starting from pushing an OpenFlow rule that enables VLAN tagging until the first packet leaves the switch with a VLAN tag. Except OVS, the rule table population strategies affect each investigated devices differently. OVS behaves the same for all four strategies with a median delay of 1.27 ms to 1.31 ms. The approach used in OVS is described in [12] and uses atomic rules which avoid overlapping rules at all and therefore reduce the effects of priorities and population order to the rule update delay.

For the NEC, the delay varies between 0.85 ms and 1.65 ms. For the Pica8 devices we observe a delay of about 2 ms for an empty flow table. The delay increases for 1000 pre-installed rules with increasing priority. Here we observe a median delay of up to 34 ms and 45 ms for the two devices. This confirms the results from previous work that the population strategy is important. With the same priority the reconfiguration times are lowest while for the increasing priority the reconfiguration times increase. In general the variance of the configuration times using OpenFlow is much higher compared to the legacy reconfiguration through management commands.

III. Methodology

In this section we introduce a methodology, metrics and notations for describing the reconfigurability of partial SDN deployments. We first discuss the system model. Based on the system model we show how the maximum global reconfiguration rate of a network topology is calculated. Besides the graph properties of the topology, the placement of the legacy and SDN devices in partial deployments is important for the global reconfiguration rate. We discuss a best case, worst case and random placement of legacy and SDN devices in the network. Afterwards we discuss the maximum feasible reconfiguration rate of all possible placements in general and of a specific topology in particular as an example. At the end of this section we introduce the potential P, an intuitive metric for comparing different topologies in terms of their expected gain in reconfigurability when deploying SDN.

A. System Model

In our model the reconfigurations arrive through the NSAL in a probabilistic manner following a Poisson process. The inter-arrival times are negative exponentially distributed. This

Table I: Key variables and notations used in the paper.

Notation	Description
λ_g	Global arrival rate
$\lambda(v)$	Arrival rate at device v
V	All nodes/devices in the topology
$h(v)$	Deterministic configuration time of device v in seconds
$\lvert V \rvert$	Number of devices in the topology
$\sigma(s, t\|v)$	Boolean indicating path from device s to t includes v
$c_B(v)$	Centrality of device v
h^L, h^S	Reconfiguration time of legacy and SDN devices

models the combined request stream from different network applications such as load balancers or firewalls on top of the NSAL. The NSAL translates the reconfigurations to configuration commands or OpenFlow messages for each device and forwards them to the devices. Different assumptions can be made in the way how the commands are completed on the device. We assume a FIFO queue of all the commands for each node. For legacy devices this might not be a queue per device, but rather a blocking TCP management connection which requires the NSAL to store the commands until the device is free again to accept the next command. Regarding SDN devices, the configuration times are reported to be limited by the interaction between CPU and ASIC [13]. As a result the commands are queued by the switch software running on the CPU, e.g. the OpenFlow agent.

The measurements show that the command execution time on the devices depends on several influence factors. For our model we assume a common deterministic reconfiguration time for all legacy devices and likewise for all SDN devices (h^L and h^S). One device is therefore modeled as an M/D/1 system following Kendall's notation.

Figure 5 shows the arrival process at the NSAL with the global reconfiguration rate λ_g. The reconfigurations are then translated to device specific configuration commands and placed in the queues on the devices. Therefore, multiple reconfigurations can be performed in the network at the same time, if they have no common device. On the other hand if they share a device, the configuration commands of the blocking devices are enqueued.

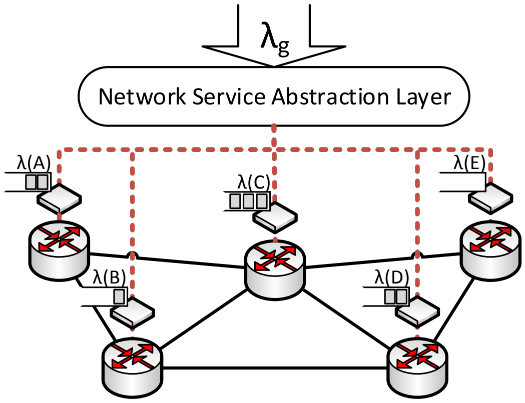

Figure 5: Configuration model with one configuration queue per device showing local and global configuration arrival rates

The maximum achievable reconfiguration rate can be computed using queuing theory. Each device v has the deterministic reconfiguration time $h(v)$. Not every global reconfiguration includes every device, so only a fraction of λ_g has to be processed by one device which leads to a local arrival rate $\lambda(v)$. The fraction of the global reconfigurations processed by a single device can be chosen in different ways. For example the fraction for each device can be deduced from historic data. Or it can be estimated based on the intended use of the partial SDN-deployed network at hand. We choose the fractions based on the transition use case of setting up VLAN tunnels in the network. If the tunnel endpoints are chosen randomly from all devices in the network and all tunnels are setup using the shortest path, the probability of device v being part of one tunnel is exactly the betweenness centrality $c_B(v)$. Besides the use case of tunnel configuration, the betweenness centrality follows the intuition that a more central device is part of more reconfigurations in the network. $c_B(v)$ of a specific device v is defined in [14] and specifies the fraction of all shortest paths which include the specific device:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

Here $\sigma(s,t|v)$ equals one if the path from device s to device t includes v otherwise zero. The local configuration rate can then be computed as:

$$\lambda(v) = c_B(v) \cdot \lambda_g$$

For a given topologie and $h(v)$, it must hold true:

$$\lambda(v) \cdot h(v) < 1 \forall v$$

or else the configuration queue grows unrestricted. This leads to a maximum global configuration rate of a given network:

$$\lambda_{g,max} = \max_{v \in V} \left(\frac{1}{c_B(v) \cdot h(v)} \right)$$

B. Network Realizations

The measurements show that SDN devices support a higher configuration rate compared to legacy devices. However, it is not enough to just consider the number of deployed SDN devices, but also their placement in the topology, i.e. which of the legacy devices in the current network are replaced by new SDN devices. For a given deployment rate and topology, the SDN devices can be placed in different ways. We call the resulting networks different realizations of the same topology. For example for a device count of 9 in a topology and a deployment ratio of 22 %, there are $\binom{9}{2} = 36$ possible network realizations, including the best and worst case realizations. Two possible realizations, best case and worst case, are shown in Figure 6. In the best case the fast SDN devices are placed most central (left in the figure), in contrast in the worst case (right in the figure) they are placed on the edge of the network. Best case and worst case result in largely different feasible configuration rates. For a deployment rate of 22 % and $h^L = 650\,ms$ and $h^S = 1\,ms$, the best case realization

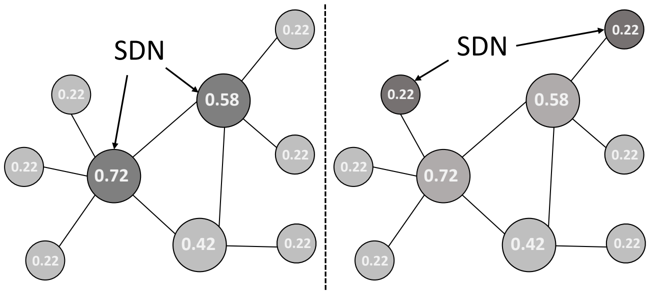

Figure 6: Best case (left) and one worst case realization (right) of a topology with 9 nodes and a deployment ratio of $\frac{2}{9}$. The annotations in the nodes denote in-betweenness centralities.

has a maximum rate of $1389\,s^{-1}$, while in the worst case the network can only process 2.14 reconfigurations per second.

The maximum possible reconfiguration rates of all possible network realizations can be summarized in the following way: We assume that the reconfiguration bottleneck is always a legacy node. This is true for many topologies if the SDN reconfiguration rate is much faster than the legacy reconfiguration rate. Therefore, regarding the maximum global reconfiguration rate, the most central non-SDN device constrains the reconfiguration rate.

The number of network realizations with a specific device as bottleneck can be deduced from counting the placement combinations. The maximum reconfiguration rate of one device, if the device is legacy, is given with:

$$\lambda^L_{max,v} = \frac{1}{h^L \cdot c_B(v)}$$

We sort all rates and index the resulting set with index x:

$$\Lambda_{max} = \{\lambda^L_{max,v}(x)\forall v \mid \lambda^L_{max,v}(x) \leq \lambda^L_{max,w}(x+1)\}$$

With this we can compute the fraction of all possible topology realizations which are constrained by the maximum reconfiguration rate at index x of Λ_{max} with l legacy devices out of $|V|$ total devices in the topology:

$$\frac{\binom{|V|-x}{l-x}}{\binom{|V|}{l}}$$

By applying the formula for all $\lambda_{max,v}$ in the set Λ_{max} we get the maximum reconfiguration rate distribution of all realizations. Next we discuss how the deployment ratio relates to λ_{max} and furthermore we apply the described placement counting to an example topology.

C. Feasibility of Reconfiguration Rates

We define a reconfiguration rate λ_g feasible for a given SDN deployment ratio and network realization if the rate is less than $\lambda_{g,max}$. Therefore when only considering the SDN deployment ratio, there exists a range of feasible configuration rates. Depending on the deployment ratio we can identify three feasibility regions for a given network topology and given reconfiguration times (h^L and h^S). Figure 7 qualitatively illustrates the three feasibility regions.

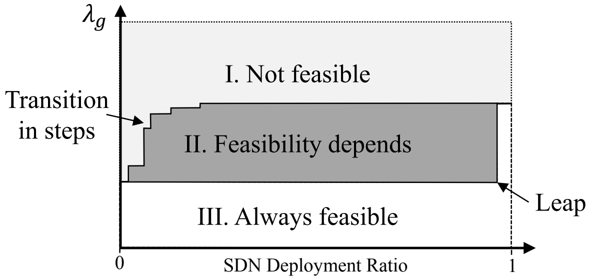

Figure 7: Qualitative feasibility regions depending on the SDN deployment ratio and maximum possible reconfiguration rate over all possible network realizations.

I. All reconfiguration rates in the region above the best case realization are infeasible, independent of the SDN and legacy device placement
II. The reconfiguration rate is feasible if the different switch types are placed such that $\lambda_g < \lambda_{g,max}$ holds true.
III. Even for the worst case realization the requested reconfiguration rate is feasible.

For a network without any SDN devices, there is only one network realization and therefore $\lambda^{WC}_{g,max}$ equals $\lambda^{BC}_{g,max}$ and there is no feasibility region II. With growing SDN deployment more and more central switches can be SDN enabled and the best case rate grows quickly until all devices in the network's core are replaced by SDN switches and the growths saturates. The border between region II and III where the worst case realizations are located has a contrary behavior and first stays constant until it leaps to $\lambda^{BC}_{g,max}$ when $100\,\%$ deployment is reached and there is only one possible realization again.

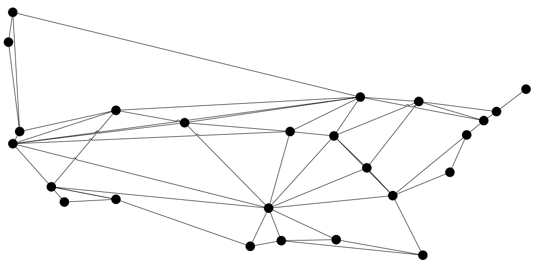

Figure 8: AT&T MPLS topology (Internet Topology Zoo [15])

Next we take a closer look at feasibility region II for a real-world topology. The goal is to better understand the expected benefit of SDN deployment, if the deployment strategy is not optimal, e.g. a semi-random upgrade strategy. We evaluate the wide area network shown in Figure 8, the AT&T MPLS topology taken from the Internet Topology Zoo [15].

For the evaluation of the expected gain we use the configuration delays of the NEC device, as the NEC is the most recent carrier-grade hardware in the testbed with legacy and SDN mode. Hence we set $h^L = 650\,ms$ and $h^S = 1\,ms$. The topology has a device count of 25 and there are 2^{25} possible network realizations in total.

Figure 9 visualizes the maximum reconfiguration rate for all possible network realizations of the AT&T MPLS topology.

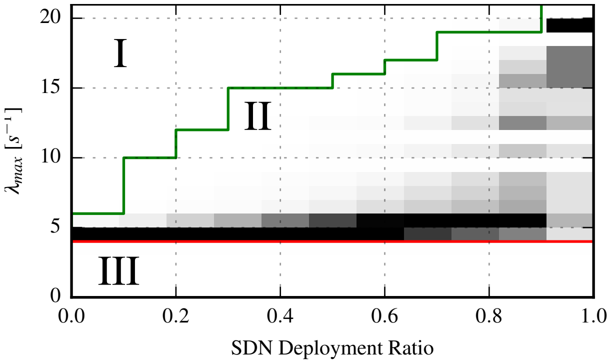

Figure 9: Analytical solution space for the AT&T topology for deployment ratios between 0 and 1 in steps of 0.1. λ_{max} is quantized to steps of 1. Color density indicates where the majority of network realizations are located. Green and red show the best case and the worst case realizations, respectively.

The x-axis denotes the SDN deployment ratio. The y-axis denotes the maximum global reconfiguration rate. The red line on the bottom indicates the worst case network realization. The green line indicates the maximum reconfiguration rate for the best case. We can observe that for this topology the border between feasibility area II and III is constant at $\lambda_{g,max} = 4.16\,s^{-1}$. The I/II border grows steadily for an increasing SDN deployment ratio.

The figure also shows the number of possible network realizations in area II using density indications. For the density indications we quantize $\lambda_{g,max}$ and the SDN deployment ratio to steps of 1 and 0.1, respectively. It can be observed that until a ratio of ≈ 0.8 is reached, most deployments are close to the worst case w.r.t $\lambda_{g,max}$. This is because a single slow device at a central position can constrain the global reconfiguration rate and the probability for this is high in a random SDN deployment. From this it follows that the transition strategy has to carefully select which devices to upgrade, as a random replacement of legacy devices with SDN devices will most likely not benefit the reconfigurability of the network.

D. Potential

In order to evaluate and compare topologies in terms of their expected increase in global reconfiguration rate when deploying SDN, we propose the *potential* P of a topology. The feasibility regions show that the centralities are the key factor to the potential reconfigurability gain. Therefore P is independent of the configuration times of the devices and focuses on centrality c_B of each node. P is defined as the mean relative positive squared deviation from the median of the node centralities. C_{med} is defined as the median of all node centralities, $C_{med} = median\{c_B(v), \forall v \in V\}$. C_+ is the set of all node centralities greater than the median, $C_+ = \{h_B(v), \forall v \in V | h_B(v) > C_{med}\}$.

$$P = \frac{\sum_{c}^{\forall c \in C_+} (\frac{c - C_{med}}{C_{med}})^2}{|C_+|}$$

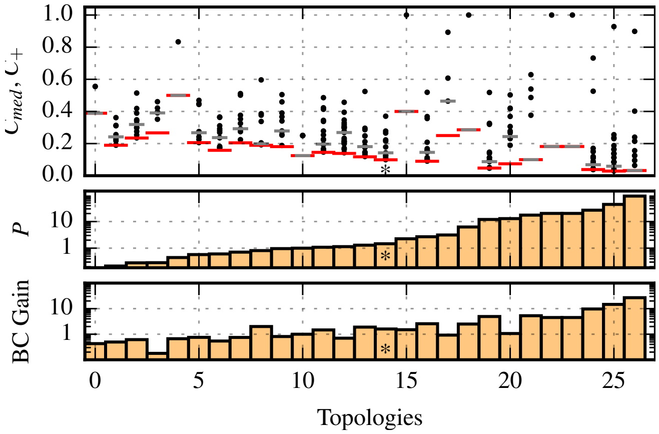

Figure 10: Centrality distribution, potential and best-case gain for a deployment ratio of 25 % of a random subset of the Internet topology zoo. C_{med} is shown as red horizontal line. C_+ as black dots. As bigger the distance between maximum and median is, the potential for increasing reconfiguration rate and BC gain by SDN deployment is.

With the deviation from the median, the potential P puts an emphasis on the beginning of the migration phase with 0 % to 50 % of the devices being replaced by SDN devices. In general, the metric favors topologies with a high number of nodes with a high centrality and decreases for topologies without central nodes.

IV. Topology Investigation

In the following we evaluate the achievable gain in terms of global reconfiguration rate when deploying SDN in real-world networks. We do so by applying our methodology to the Internet topology zoo. The Internet topology zoo provides a database of 200 to 300 publicly available real-world topologies ranging from small testbed installations to large global-scale backbone networks. We filter the topology zoo by discarding topologies which are not connected or not unique in terms of node centralities. After filtering, our topology set for the evaluation contains 81 topologies with a node count between 4 and 74 and an average of 22 nodes per topology.

In the best case, to increase the maximum reconfiguration rate of the network, an operator would choose to upgrade devices to SDN which are central or receive the most reconfiguration requests and are slow to reconfigure at present. Therefore, we define the best case (BC) gain as the ratio between the achievable reconfiguration rate of a topology with and without SDN devices at the nodes in the topology graph with the highest centralities. For example, a BC gain of 1 refers to a doubling of the maximum achievable reconfiguration rate.

Next, we first discuss a random subset of the topologies in greater detail. Afterwards we summarize the findings of all topologies. At the end, we discuss the relationship of the gain and the potential P based on the topology set. Reconfiguration times are set to $h^L = 650$ and $h^S = 1$.

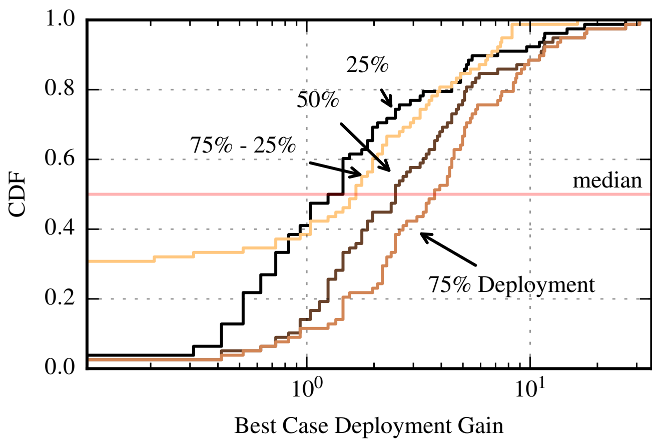

Figure 11: Best case deployment gain over 81 real world topologies for deployment ratios of 25 %, 50 %, 75 % and the difference in gain between 75 % and 25 %. A deployment gain of 1 translates to a doubling of the maximum global reconfiguration rate of the network.

Figure 10 illustrates the BC gain, the potential P and the centralities for 26 of the 81 topologies. The BC gain is given for a deployment ratio of 25%. P and BC gain are shown on a logarithmic scale and the topologies are sorted by P. The upper part of the figure shows the median of the centralities C_{med} with a wide red bar. The shorter gray bar shows the 75 % quartile, i.e. the most utilized legacy node in a 25 % SDN deployment. The dots indicate the centralities greater than the median (C_+) of each topology. The AT&T MPLS topology from Figure 9 is located at position 14 and marked with a star.

Two conclusions can be drawn from the figure. First, the potential P gives a good estimate of how well topologies can profit from SDN deployment compared to each other. However there are exceptions like topology 3, 17 and 20 which can profit far less from the SDN deployment than P suggests. For example in comparison to topology 2 and 4, the difference between topology 3 most central node (upper most dot) and the 75 % quartile (gray short bar) is smaller. Hence, topology 3 can not profit as much from an 25 % SDN deployment as topology 2 and 4. Second, the achievable best case gain for 25 % deployment varies greatly between different topologies. For topology 3 we observe a gain of only 0.18, whereas topology 26 exhibits a gain of about 27. The AT&T MPLS topology is with a gain of 1.6 close to the median of 1.5 of all shown topologies. On average, there is a gain of 3.4.

Next, we evaluate the achievable gain for all topologies in the topology set for 25 %, 50 % and 75 % SDN deployment. Figure 11 shows the cumulative distribution function (CDF) of the best case deployment gain on a logarithmic scale. "75 % - 25 %" denotes the difference between the 25 % and 75 % gain for each topology. The points where 50 % of the topologies have less than a specific gain are highlighted by the red semi-transparent vertical line. The line intersects the CDFs at a gain of 1.37, 1.67, 2.57 and 3.76 from the left to the right.

	h^L/h^S 5	10	100	650	1000
25%	0.66	0.85	0.94	0.94	0.94
50%	0.48	0.71	0.80	0.80	0.80
75%	0.35	0.56	0.74	0.74	0.74

Table II: Pearson correlation between potential P and BC gain for 25 %, 50 % and 75 % SDN deployment and four different ratios of h^L/h^S with $h^L = 1$.

The figures shows that for half of the real-world topologies, upgrading half of the nodes to SDN results in the best case in an up to three to four times higher achievable reconfiguration rate. If only one forth of the devices are upgraded, the gain drops to two to three times in the best case. If three forth of the devices are upgraded, the achievable reconfiguration rate increases to four to five times. The CDF for the difference between 75 % and 25 % deployment answers the question of how much gain can be expected when continuing SDN deployment from 25 % to 75 % of the devices. The results show that for half of the topologies, an continuation in the upgrade process increases the reconfiguration rate up to two to three times, similar to the upgrade from all-legacy to 25 %.

Next, we discuss the correlation between the best case gain and the potential based on the investigated topology set. Table II summarizes the Pearson linear-correlation between the best case gain and the potential P for three different deployment ratios and five different ratios between the configuration times of the legacy and SDN devices (h^L/h^S). In general, we observe that the linear correlation is stronger for higher values of the ratio h^L/h^S. This comes from the fact that P expects the slowest SDN device to be faster than the fastest legacy device, which is less likely for lower ratios of h^L and h^S. Furthermore, the correlation decreases for higher deployment ratios. This is not surprising as P is defined to favor lower deployment ratios. The Spearman rank-correlation (not shown in the table) is ≥ 0.90 for all combinations of 25 % and 50 % deployment and all configuration time ratios shown in the table and ranges between 0.35 and 0.75 for 75 % SDN deployment. From this it follows that the potential P is both suitable for comparing the gain of topologies in magnitude and rank.

V. Summary and Discussion

In this section we summarize and discuss the presented measurements, methodology, metric and the results of the topology investigation. In Section II we conduct detailed measurements of OpenFlow-enabled and legacy devices. The results there show that the evaluated legacy devices have highly varying reconfiguration times from 3 ms to 20 ms and up to 650 ms. The results for the OpenFlow-enabled devices show that the reconfiguration time also highly depends on the device type and additionally on the amount of previously installed flow rules. Here we observe reconfiguration times from 1 ms to 100 ms. Based on the measurements we choose the configuration times of the only device in our study with OpenFlow and legacy mode for the remainder of the paper. Hence, we set legacy reconfiguration time to 650 ms and SDN reconfiguration time to 1 ms. This is at the same time a best case choice from the perspective of the upgrade process. An upgrade of a device eliminates it in most cases as a potential reconfiguration bottleneck in the network.

In Section III we first discuss the system model. We argue that configuration tasks are formulated centrally, e.g. by a network management application on top of an abstraction layer, and afterwards placed in device specific queues. Based on previous work on SDN migration we introduce a scenario where each device receives a fraction of the global reconfigurations generated by the abstraction layer using the device's betweenness centrality as a measure for this fraction. However, the presented methodology is not limited to the in-betweenness centrality as workload metric for a device. Other options could be to determine the workload of a device based on historic data. Afterwards we introduce the concepts of feasibility and maximum achievable global reconfiguration rate. For a specific topology in a specific migration stage, a global reconfiguration rate is feasible if no single device is overloaded by its fraction of the reconfigurations.

Subsequently we discuss how the distribution of SDN and legacy devices in the network influences the maximum achievable rate. Here we see that there is a best case strategy which replaces first the devices with the highest workload and a worst case strategy to replace first all edge devices with low workload. Additionally we present how the maximum reconfiguration rate of all possible upgrade strategies can be calculated. Based on an example we deduce that a random upgrade strategy results much more likely in a maximum reconfiguration rate close to the worst case than to the best case. This stems from the fact that a single slow device can severely impact the global reconfiguration rate and that the number of devices with low centrality is much higher compared to the number of devices with high centrality. At the end of the section we introduce a metric which captures how well a given topology responds to the best case migration strategy. As we see in the evaluation the metric can capture the best case gain well up to a 50 % deployment ratio.

In Section IV we apply the methodology to a large number of real-world topologies from the Internet topology zoo. We evaluate the expected best case gain for deployment ratios of 25 %, 50 % and 75 %. The results show that for 50 % of the topologies an SDN deployment ratio of 25 % increases the global reconfiguration rate up to two to three times. Furthermore, we see a similar increase when continuing the migration from 25 % to 75 %. Hence, when three out of four devices are upgraded we observe the global reconfiguration rate being up to 5 times compared to the all-legacy case. In general it is surprising that the 650-fold increase in terms of possible reconfiguration rate of the SDN devices results in a relatively low increase in the global reconfiguration rate even in the best case.

In conclusion, an operator can apply the following three step approach to determine the reconfigurability of his network. In a first step, the operator measures or requests from the vendor the reconfiguration times of the devices ($h(v)$) in the network depending on the foreseen use case, e.g. for VLAN tagging. As a second step, the operator determines for each node in this network the fraction of reconfiguration it receives, either by monitoring the currently deployed network, or by approximating the future transition scenario through other means, such as a simulation. Subsequently, the operator can apply the presented methodology to determine a suitable upgrade strategy with respect to the desired maximum global reconfiguration rate.

VI. Related Work

Generally it is expected that the transition to SDN networks will lead over mixed SDN/legacy deployments and therefore some work has been done to integrate legacy control and management cases into SDN networks [4], [16], [5], [6], [17]. Although in most works the focus is not on the rate of reconfigurations but on the integration techniques.

One way of integration is the usage of ”fake” legacy-routing messages (e.g. OSPF messages) to cause the desired behavior of the legacy devices [4]. The authors show a network architecture which enables traffic steering in hybrid networks along any path in the network with per-destination granularity. This approach is therefore well in line with our model, although the reconfiguration times might differ. Closely related are approaches that try to mimic SDN behavior on legacy devices e.g. with ACL rules [5]

Another approach is to use legacy ACL rules to ensure the traversing of an SDN device. HybridFlow [17] is a prototype that implements this approach and provides a combined North-bound API to the network applications. A prototype shows the feasibility of a combined NSAL, though the authors of [17] do not consider the differences regarding configuration rates.

Panopticon [6] implements a similar approach using VLAN tunnels for legacy/SDN configurations. The results show that only a small share of nodes has to be upgraded to provide full SDN capabilities for the network. Additionally, a deployment strategy based on the switch egress volume is shown.

Another research angle is to reduce the number of necessary reconfigurations, this can be done for SDN using wildcard rules. Authors of [18], [19] aim to reduce the control plane load by limiting the network state view of the centralized controller. As a side effect this also reduces the necessary reconfiguration rate and attenuates the issues we solve here.

Differences in the time to update switches can also cause congestion and inconsistencies in a network in general. Authors of [9] provide an algorithm that schedules network updates efficiently. The focus of the work is on optimizing rule ordering for network updates which are non-atomic in order to improve reconfiguration speed.

Complementary to our observations regarding the importance of centrality w.r.t. to reconfigurations, authors of [20] evaluate gradual deployments of SDN devices in terms of throughput. They review real world topologies, where some percentage of the devices are replaced. They observe that for traffic engineering a relatively small deployment of about 20%

can reduce congestion and maximum link usage in real ISP and enterprise topologies effectively.

There are several works providing measurements of SDN switches and also control plane delays. E.g. [13], or [21], [11]. The provided measurements study the installations of batches of rules and not the installation of a single rule as in our scenario. In general the results in literature strongly support our results and show usually delays in the range from 1 ms to 10 ms. This leads to the conclusion that poor performance is one of the main scalability issues in SDN networks [22].

Authors of [23], [24] provide a traffic model of OpenFlow. However, the authors focus on modeling a reactive SDN, while we aim to model the maximum achievable reconfiguration rate independent of the mode, reactive or proactive, and especially for mixed SDN/legacy networks.

VII. Conclusion and Outlook

Resulting from economical and technical reasons, there will be a transition period where SDN-enabled and legacy devices with often slow configuration interfaces have to co-exist in the same network segment. Related work already introduces concepts how such a hybrid network can be operated, for example by using VLAN tags to implement forwarding decisions on the legacy devices. However, it is not investigated how the presence of slow (in terms of reconfiguration rate) legacy devices affect the achievable global reconfiguration rate. In this paper we investigate the impact of different SDN deployment ratios on the maximum reconfiguration rate of the network.

In general the results show that even a small number of inflexible legacy devices severely reduce the maximum reconfiguration rate of the network. Even at higher deployment ratios, the global reconfiguration rate does not increase by more than factor five compared to the all-legacy deployment. Therefore the results raise the question how advanced SDN scenarios, which require timely reconfigurations, can be realized in networks with only partial SDN deployment. This is a gap in current research and has to be investigated further to enable a smooth and beneficial migration phase of existing networks to SDN-enabled networks. Future work in this area should extend the number of measured devices to generate an abstract model of the reconfiguration times of different devices and explore the possibility of configuration load-balancing to circumvent slow configuration interfaces of legacy devices.

VIII. Acknowledgment

This work has been partially funded by the German Research Foundation (DFG) under the grant number KE1863/6-1 and by the German Federal Ministry for Research and Education (BMBF) in the framework of the SARDINE project (Project ID 16KIS0261). The authors alone are responsible for the content of the paper.

References

[1] D. Kreutz, F. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," 2014.

[2] A. Doria, J. H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification," RFC 5810 (Proposed Standard), Internet Engineering Task Force, Mar. 2010.

[3] C. Sieber, A. Blenk, A. Basta, D. Hock, and W. Kellerer, "Towards a Programmable Management Plane for SDN and Legacy Networks"," in *2nd IEEE Conference on Network Softwarization (NetSoft)*, 2016.

[4] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford, "Central control over distributed routing," in *ACM Conference on Special Interest Group on Data Communication (SIGCOMM)*, 2015.

[5] R. Hand and E. Keller, "Closedflow: Openflow-like control over proprietary devices," in *Third workshop on Hot topics in software defined networking (HotSDN)*. ACM, 2014.

[6] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann, "Panopticon: Reaping the Benefits of Incremental SDN Deployment in Enterprise Networks." in *USENIX Annual Technical Conference*, 2014.

[7] J. Medved, R. Varga, A. Tkacik, and K. Gray, "Opendaylight: Towards a model-driven SDN controller architecture," in *15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2014.

[8] "OpenStack." [Online]. Available: https://www.openstack.org

[9] X. Jin, H. H. Liu, R. Gandhi, S. Kandula, R. Mahajan, M. Zhang, J. Rexford, and R. Wattenhofer, "Dynamic scheduling of network updates," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, 2014, pp. 539–550.

[10] "libfluid - The ONF OpenFlow driver." [Online]. Available: http://opennetworkingfoundation.github.io/libfluid/

[11] M. Kuźniar, P. Perešíni, and D. Kostić, "What you need to know about SDN flow tables," in *International Conference on Passive and Active Network Measurement (PAM)*, 2015.

[12] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar *et al.*, "The Design and Implementation of Open vSwitch," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2015.

[13] K. He, J. Khalid, A. Gember-Jacobson, S. Das, C. Prakash, A. Akella, L. E. Li, and M. Thottan, "Measuring control plane latency in SDN-enabled switches," in *1st ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR)*, 2015.

[14] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[15] "The internet topology zoo." [Online]. Available: www.topology-zoo.org

[16] C. Jin, C. Lumezanu, Q. Xu, Z.-L. Zhang, and G. Jiang, "Telekinesis: controlling legacy switch routing with openflow in hybrid networks," in *1st ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR)*, 2015.

[17] S. Huang, J. Zhao, and X. Wang, "Hybridflow: A lightweight control plane for hybrid sdn in enterprise networks," in *IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, 2016.

[18] 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, 2011, pp. 254–265.

[19] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with difane," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 351–362, 2011.

[20] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental Deployment of SDN in Hybrid Enterprise and ISP Networks," *2nd ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR)*, 2016.

[21] A. Lazaris, D. Tahara, X. Huang, E. Li, A. Voellmy, Y. R. Yang, and M. Yu, "Tango: Simplifying sdn control with automatic switch property inference, abstraction, and optimization," in *10th ACM International on Conference on emerging Networking Experiments and Technologies (CoNEXT)*. ACM, 2014, pp. 199–212.

[22] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *Communications magazine, IEEE*, vol. 51, no. 2, pp. 136–141, 2013.

[23] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "Modelling of OpenFlow-based software-defined networks: the multiple node case," *Networks, IET*, vol. 4, no. 5, pp. 278–284, 2015.

[24] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *23rd International Teletraffic Congress (ITC)*, 2011.