Investigating Isolation between Virtual Networks in Case of Congestion for a Pronto 3290 Switch

Anh Nguyen-Ngoc*, Stanislav Lange*, Steffen Gebert*, Thomas Zinner*, Phuoc Tran-Gia*, Michael Jarschel§
*Institute of Computer Science University of Würzburg, Germany.
Email: {anh.nguyen,stanislav.lange,steffen.gebert,zinner,trangia}@informatik.uni-wuerzburg.de
§ Nokia, München, Germany.
Email: michael.jarschel@nsn.com

***Abstract*—Performance isolation between virtual resources is one of the key features of network virtualization. It is typically realized by configuring queues with specific rate guarantees on the egress ports of network devices. The drawback of this architectural choice, however, is that traffic from several ingress ports may result in congestion on an egress port. Hence, the question arises to which extent isolation between virtual networks is realized in state-of-the-art hardware. This work aims at investigating whether congestion within one virtual network may affect the throughput performance of another virtual network. For that, measurements in a local testbed using a Pronto 3290 switch running an OpenFlow-enabling Pica8 firmware are performed.**

I. Introduction

Network Virtualization (NV) is a networking paradigm which allows overcoming the deficiencies of the current network architecture like ossification of the protocol stack and missing innovation [1], [2]. NV allows the operation of multiple logical or virtual networks which are isolated from each other on a common physical substrate. These virtual networks can be instantiated with their own application-specific naming, topology, routing, and resource management. Hence, network virtualization is seen as an important building block for future networks as proposed by many Future Internet initiatives and researchers [1]–[3].

An important feature of virtual networks is to ideally completely isolate each network, while they still share the same common physical resources. The isolation is typically realized by configuring buffers, rate limits, and queueing disciplines on incoming or outgoing interfaces. Hence, additional delays and jitter may be introduced due to waiting times in buffers at the switching fabric, the incoming, or the outgoing interface [4]. In case of a First-In-First-Out queueing discipline, buffering at the ingress results in head-of-line blocking and reduced throughput [5]. To overcome this issue, virtual output queues at the ingress have been proposed and implemented in order to achieve a high throughput of approximately 100% [6].

Explicit rate limits or guarantees for specific virtual networks, however, are typically configured at the outgoing queue. Although the crossbar fabric of a switch meets the Clos condition and is thus non-blocking, congestion may happen due to several input ports sending to the same output port. Accordingly, the question arises, whether congestion in one virtual network, realized by rate guarantees on an outgoing interface, may also influence the performance of other virtual networks. Particularly, the impact of several parameters like the specific configuration, the duration of the congestion, or the intensity of the overload in one virtual network may have an impact on the resource isolation and on the throughput of the other virtual networks.

The authors of [7] present a theoretical model of a multi-core router that supports multiple VNs simultaneously and compare it with a simple store-and-forward router model that uses a single processor for all incoming traffic. Due to isolation as well as VN-specific scheduling and processing of packets, the multi-core model achieves better performance with respect to delay and jitter when load is increased for all VNs. However, the focus of this work lies on the resilience aspect of isolation mechanisms, especially in the presence of sudden bursts that congest the underlying physical queue of the multi-core router.

In [8], the behavior of virtual routers that are based on commodity hardware is investigated. For this, competing objectives including performance, isolation, flexibility, and fairness are taken into account. Problems of software-based traffic classification approaches that use a single physical queue are illustrated in an analysis of their isolation performance. Emulating a system with multiple physical queues via a switch that is connected upstream of multiple NICs and comparing it to a system that uses software classification shows that with increasing load, the latter fails to maintain isolation. This results in unexpected behavior on all virtual queues.

This paper works towards an understanding of the questions raised regarding isolation performance by investigating key metrics like the packet loss per virtual network using measurements in a dedicated testbed. The testbed is comprised of several traffic generators and a Pronto 3290 switch. Measurements for different switch configurations and loads are conducted and results are highlighted.

In addition, an impact of the IEEE 802.3X Ethernet flow control [9] on the isolation is observed. IEEE 802.3X is a mechanism which allows a network device to send an Ethernet frame, called Pause frame, to tell its neighbor that it is experiencing overload, e.g., when the device is receiving data faster than it can handle. Pause frames signal the device at the other end of the link to stop transmission for a certain period of time. When it receives the Pause frame, the sender should stop the traffic flow for a certain duration which is specified in the frame, and buffer the packets until the receiver is ready to

978-1-4799-5804-7/15/$31.00 2015 IEEE

accept them again. The purpose of flow control is to prevent packet loss by handling input buffer congestion. When flow control is disabled, the overloaded device will drop packets.

The paper is structured as follows. Section II provides an overview of the hardware and software components utilized in the testbed as well as the experiment parameters and scenarios. Subsequently, Section III presents results of multiple experiment repetitions with different system configurations. Finally, Section IV concludes the paper and gives an outlook on future research directions.

II. Methodology

In this section, the measurement setup, including the used hardware and software is described. Additionally, an overview of the performed experiments is provided alongside the notation for adjustable parameters and key performance indicators. Finally, the various experiment scenarios in terms of parameter values and hardware configuration are discussed.

A. Measurement Setup and Configuration

In order to evaluate the isolation performance of the outgoing physical queue of a switch that contains multiple virtual queues, a testbed has been set up according to Figure 1.

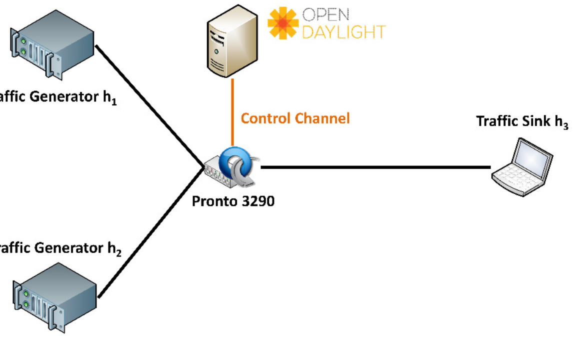


Fig. 1: Testbed Setup

Its main components are two traffic generating hosts, h_1 and h_2, which are connected to the Pronto 3290 OpenFlow switch and send Iperf[1] UDP traffic to a third host, h_3, which is also connected to the switch. The switch runs PicOS[2] version 2.0.14 which comes with OpenFlow version 1.0 alongside Open vSwitch 1.10.0. For its OpenFlow functionality, the switch requires a connection to an OpenFlow controller. In this work, the OpenDaylight controller[3] was chosen. In this environment, separate virtual queues are established on the switch for each individual sender. Thus, traffic originating from a specific source is directed to a specific virtual queue on the switch's output port via statically preconfigured OpenFlow rules. Virtual queues on the switch are configured according to the documentation available on the Pica8 website[4]. This is achieved via the `ovs-vsctl` command which is used for setting up queues with minimum and maximum rates on the outgoing port of the switch.

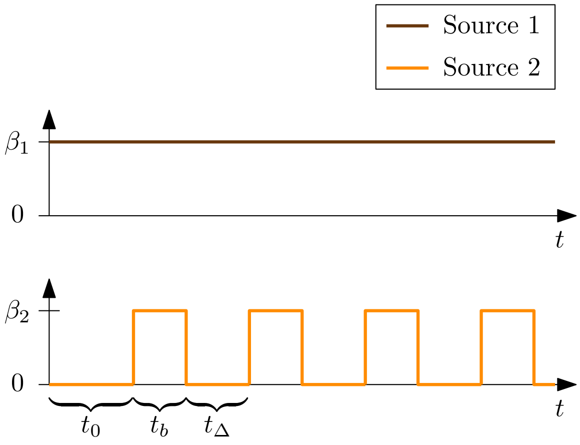


Fig. 2: Experiment scheme for $n_b = 4$

B. Course of Experiments

For the actual evaluation, several influence parameters and performance indicators have been identified and were integrated into an experiment scheme. The basic scenario is depicted in Figure 2. It consists of traffic generator h_1 sending UDP packets with a constant rate of β_1 over the switch's first queue and the second traffic generator sending regular bursts of UDP packets with a rate of β_2 over the second queue. The two queues are referred to as q_1 and q_2, respectively, and have rate limits β_1^l and β_2^l. In all experiments, the sending rate of h_1 is equal to q_1's limit, i.e., $\beta_1 = \beta_1^l$. At the beginning of each run, q_2 is idle for time t_0 in order to assure a stable system state. Then, n_b bursts of UDP traffic are produced at the second traffic generator via Iperf in regular intervals. These bursts last t_b and are interleaved with pauses of t_Δ. During the bursts, the rate limit of q_2 is exceeded, i.e., $\beta_2 > \beta_2^l$ holds.

Measured parameters include the relative and absolute packet loss on q_1, denoted as p_1 and p_1^a, respectively, as well as the received rates for both queues, β_1^r and β_2^r. These values allow statements about the influence of q_2's bursts on the performance of q_1, e.g., $p_1 > 0$ implies a violation of the isolation between queues. Furthermore, changes in the receive rate for the first flow, β_1^r, indicate a performance degradation. Hence, these parameters can be used to quantify the impact of bursts on the overall system reliability.

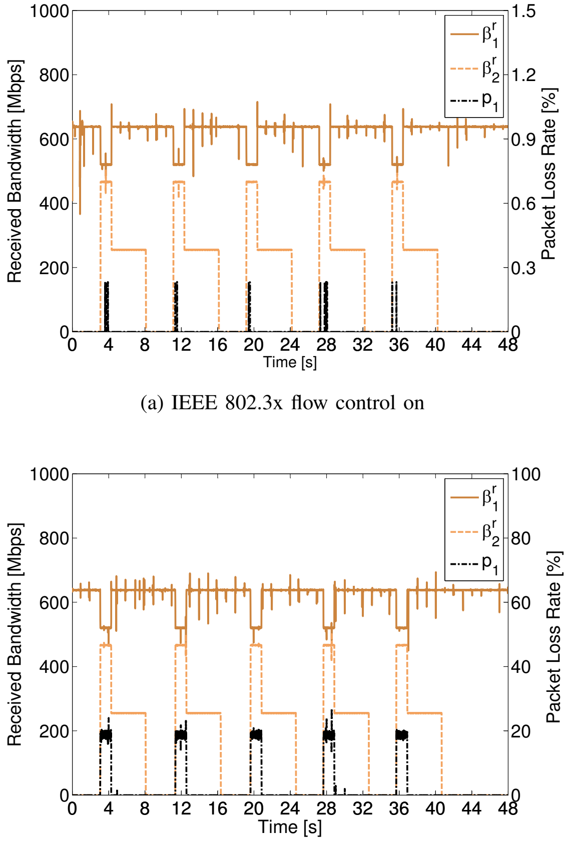
C. Investigated Scenarios

Given the total capacity c of the link between switch and traffic sink, the total load can be computed as $\rho = \frac{\beta_1+\beta_2}{c}$. This is a key influence factor as in the context of a low total load, the system is more likely to compensate short or less intense bursts. Furthermore, the amount of resources allocated by the network operator can have an impact on isolation performance. Hence, a parameter for the provisioned bandwidth, $\rho_p = \frac{\beta_1^l+\beta_2^l}{c}$, is introduced. Table I provides an

[1] http://iperf.sourceforge.net/
[2] http://www.pica8.com/open-switching/open-switching-overview.php
[3] http://www.opendaylight.org/
[4] http://www.pica8.com/document/picos-2.0.1-ovs-configuration-guide.pdf

TABLE I: Parameter sets used in this work

Name	t_b	t_Δ	n_b	c	$\beta_1 = \beta_1^l$	β_2	β_2^l	ρ	ρ_p
$\mathcal{P}_1$	$\in \{1s, 2s, 5s\}$	3s	5	1 Gbps	640 Mbps	465 Mbps	100 Mbps	1.1	0.75
$\mathcal{P}_2$							206 Mbps		0.85
$\mathcal{P}_3$							255 Mbps		0.90
$\mathcal{P}_4$						640 Mbps	100 Mbps	1.28	0.75
$\mathcal{P}_5$							206 Mbps		0.85
$\mathcal{P}_6$							255 Mbps		0.90



(a) IEEE 802.3x flow control on

(b) IEEE 802.3x flow control off

Fig. 3: Single experiment with parameter set $\mathcal{P}_3$ and $t_b = 5s$

overview of parameter combinations discussed in this work. Additionally, flow control mechanisms like IEEE 802.3X Ethernet flow control [9] can prevent overload by limiting the sending rate of traffic sources. However, this behavior can be turned off by the client in order to produce the desired amount of traffic regardless of the system state. In order to quantify the impact of flow control, experiments have been conducted for both settings.

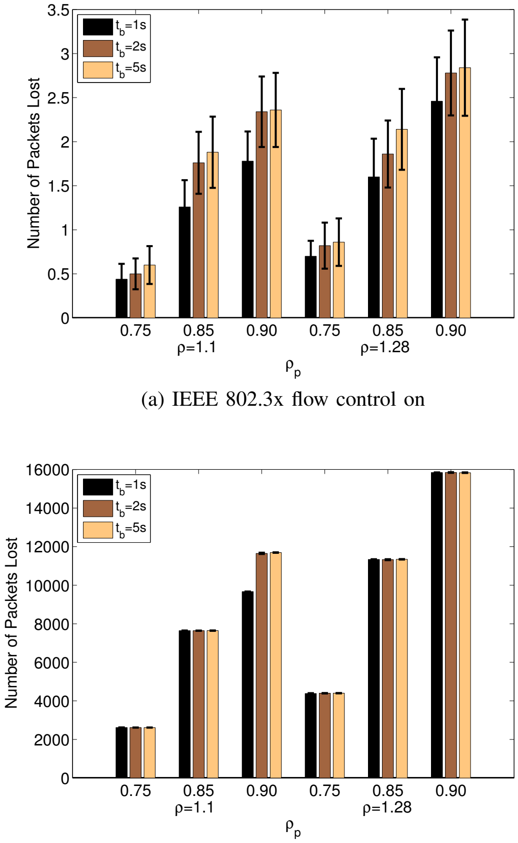
III. Results

This section presents results gained during the experiments that were described in Section II. First, results obtained from a single experiment run are shown in order to demonstrate the experiment scheme. Then, the influence of parameters like the system load ρ, the provisioned bandwidth ρ_p, and the burst duration t_b on the resulting packet loss is investigated. Finally, the relationship between these parameters and the amount of time during which packet loss occurs is analyzed.

A. Single Experiment Run

In order to provide an illustrative example of the performed measurements, the plots of Figure 3 provide a view on the data captured during a single experiment run with flow control either enabled or disabled. The x-axis indicates the time while the two y-axes show information on received throughput at the sink on the left hand side as well as the packet loss experienced by the traffic originating from the first traffic generator on the right hand side. Figure 3a illustrates the behavior in case of enabled flow control. The bursts from the second traffic generator initially affect the amount of data received from the first source as the total sending rate exceeds the link's capacity of 1 Gbps and output queues begin to congest. While this behavior in itself indicates a breach of isolation between VNs, additionally some packet loss, less than 1%, occurs for the traffic that passes q_1. However, flow control kicks in fast and restricts the sending rate of the second generator, thus avoiding further congestion and therefore packet loss or bandwidth degradation for the first flow. Furthermore, variation in β_1^r, the received rate from host h_1, can be observed. Currently, this behavior can not be explained and is subject to future investigations.

In contrast to the outlined scenario, the packet loss experienced in the absence of the flow control mechanism is orders of magnitude higher, i.e., at around 20%. Detailed measurement data is provided in Figure 3b. Although flow control is disabled, the receive rate β_2^r also drops shortly after the bursts start. Unlike in the first case, however, this is not due to an adapted β_2, but rather the switch dropping packets destined for q_2 after experiencing an incoming rate β_2 that is higher than the provisioned rate β_2^l. The plots show that performance degradation in expectedly isolated virtual networks can be achieved by overload conditions in a single VN. While flow control does affect the intensity of the resulting degradation, flow control can not completely avoid it. In addition to the flow control mechanism on the sending side, the switch also behaves in a reactive fashion when incoming rates exceed the agreed-upon limits for a certain amount of time. A further interesting effect that is observed for enabled as well as disabled Ethernet flow control is that the received rate of traffic from the first source, β_1^r, is significantly reduced during the overload period while the



(a) IEEE 802.3x flow control on

(b) IEEE 802.3x flow control off

Fig. 4: Influence of ρ, ρ_p, and t_b on p_1^a, the number of packets lost in a single burst

second flow passes through at its full sending rate β_2.

B. Impact of Experiment Parameters on Packet Loss

As motivated in Section II, this work aims at identifying the main influence factors of the phenomena observed in the previous measurements. For this purpose, experiments with a variety of system configurations have been performed ten times. Figure 4 presents the influence of the system load ρ, the provisioned bandwidth ratio ρ_p, and the burst length t_b on p_1^a, the absolute number of packets lost in a single burst event.

The first subplot shows results gained from configurations in which flow control was turned on. While the ticks on the x-axis denote ρ_p, the ratio of provisioned bandwidth and link capacity, the y-axis displays p_1^a. In addition to the bars' height indicating the mean of measured values, the whiskers show 95% confidence intervals that were obtained by repeating each experiment 10 times. Bar colors represent different values for t_b. Results are presented for two different system loads ρ. The first observation is that confidence intervals for each group of scenarios with identical ρ and ρ_p overlap, i.e., different values of t_b do not have a statistically significant impact on the amount of lost packets when the remaining parameters are fixed. The reason for this is that the flow control mechanism reacts fast enough to prevent a large packet loss, but does not affect the amount of time during which the first flow's received rate β_1^r is reduced. Thus, packet loss acts as indicator while the bandwidth impediment constitutes the actual breach in isolation. Furthermore, increasing values of ρ_p result in a higher number of lost packets p_1^a. A possible explanation for this behavior is that preconfigured rate guarantees influence the duration until the switch reacts to the overload. Thus, the time frame in which packet loss might occur becomes longer for increasing ρ_p. Hence, the number of lost packets, p_1^a, increases. The applied system load ρ, however, does not seem to have a high degree of influence on the packet loss in case of enabled flow control. This can be observed by comparing pairs of groups that have identical values of ρ_p but different loads ρ. A likely explanation for this phenomenon is that while a higher value of ρ implies a higher congestion rate, it also leads to a faster detection of the burst by the flow control mechanism. The latter limits the damage done in terms of packet loss by limiting the sending rate at h_2.

Figure 4b displays the measurements that were obtained with disabled Ethernet flow control. Like in the previous figure, an increased number of lost packets is observed for increasing values of ρ_p. In contrast to just a few lost packets when flow control is enabled, multiple thousands of packets are lost in the context of disabled flow control. Furthermore, ρ also affects the number of lost packets. However, t_b does not seem to have an influence on p_1^a even in the absence of flow control. Except in the case of the combination $\rho = 1.1$ and $\rho_p = 0.90$, the number of lost packets per burst event does not change significantly when t_b is increased. In order to shed light on this behavior, the duration in which packet loss for packets from h_1 occurs has been recorded in each experiment. This time is denoted as t^i and values for different parameter sets are shown in Table II.

TABLE II: ρ_p and resulting t^i values for different scenarios

Parameter set	ρ_p	t^i
$\mathcal{P}_1$	0.75	0.27
$\mathcal{P}_2$	0.85	0.81
$\mathcal{P}_3$	0.90	1.2
$\mathcal{P}_4$	0.75	0.25
$\mathcal{P}_5$	0.85	0.48
$\mathcal{P}_6$	0.90	0.68

Additionally, the table presents ρ_p which quantifies the total provisioned bandwidth. Only in the context of parameter set $\mathcal{P}_3$, t^i exceeds one second which explains the difference in the amount of lost packets observed in Figure 4b. As soon as $t_b > t^i$ holds, the absolute packet loss p_1^a does not change for increasing t_b.

C. Amount and Duration of Packet Loss

Plotting the measured values for t^i and ρ_p against each other shows a proportional relationship when considered for

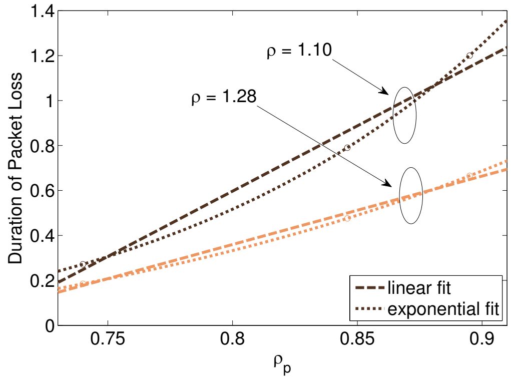


Fig. 5: Linear and exponential fits of the relationship between ρ_p and t^i for different values of ρ

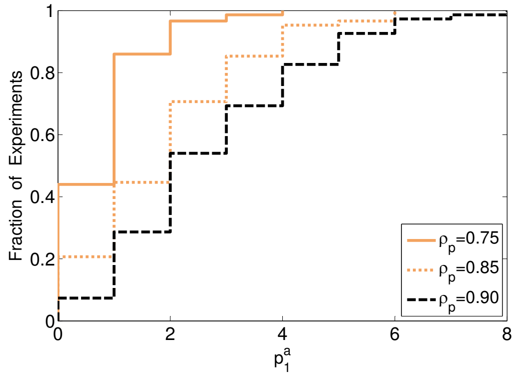


Fig. 6: Distribution of the absolute packet loss p_1^a for different values of ρ_p (IEEE 802.3x flow control is enabled)

individual loads ρ. Figure 5 illustrates the increase of t^i when ρ_p rises. Further measurements are required in order to determine the type of this relationship, e.g., linear as indicated by the dashed lines or exponential as indicated by the dotted lines. These measurements will also help explaining the internal behavior of the switch which results in different slopes for different values of ρ.

Having identified ρ_p as the main influence factor on the packet loss for the traffic from h_1 to h_3 in case of enabled flow control, Figure 6 provides an aggregated view on the empirical cumulative distribution function of p_1^a for different values of ρ_p. The x-axis denotes the number of packets lost during a single burst attempt, while the y-axis indicates the fraction of experiments for which the observed value p_1^a is less than or equal to this number. The resulting distributions are in line with the previous observations and exhibit an increasing number of lost packets for increasing values of ρ_p. For example, in the context of $\rho_p = 0.75$, p_1^a never exceeded 4 packets per burst while it did in 17% and 19% of instances for $\rho_p = 0.85$ and $\rho_p = 0.90$, respectively.

In contrast to scenarios with enabled flow control, where ρ_p is the main influence factor, multiple parameters have been identified for configurations in which the flow control mechanism is disabled. Thus, a distribution of p_1^a that solely depends on ρ_p does not provide new insights and is therefore omitted.

IV. Conclusion

In this work, the resource isolation between virtual networks in the presence of traffic bursts was investigated. Measurements performed in a testbed equipped with a Pronto 3290 switch yield numerous insights. First, even in the presence of Ethernet flow control on sending devices, a violation of the isolation between VNs is possible. Second, main influence parameters on the isolation performance were identified and investigated. It was shown that besides the overall load on the outgoing switch port also the configured rate guarantees per virtual network have a significant impact on the number of lost packets. The measurements further show a correlation between the degree of congestion on the outgoing port, the configured rate limits, and the delay until the switch reacts to the violation of the resource isolation. Future work aims at understanding the investigated behavior in detail and establishing appropriate models. These models will be verified with other switches and networking devices and adapted if necessary. Additionally, more realistic traffic patterns, e.g., TCP burstiness, will be investigated. The results and models can be used to find appropriate configurations for the involved hardware devices with respect to resource isolation.

Acknowledgment

This work has been performed in the framework of the CELTIC EUREKA project SASER-SIEGFRIED (Project ID CPP2011/2-5), and it is partly funded by the BMBF (Project ID 16BP12308). The authors alone are responsible for the content of the paper.

References

[1] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network virtualization: Implementation steps towards the future internet," in *Electronic Communications of the EASST, Volume 17: Kommunikation in Verteilten Systemen 2009*, Kassel, Germany, March 2009.

[2] N. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[3] GENI Consortium, "GENI - global environment for network innovations," 2006, information available at http://www.geni.net/.

[4] O. Hohlfeld, "Impact of buffering on quality of experience," Ph.D. dissertation, Technische Universität Berlin, 2013.

[5] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *Communications, IEEE Transactions on*, vol. 35, no. 12, pp. 1347–1356, 1987.

[6] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Communications, IEEE Transactions on*, vol. 47, no. 8, pp. 1260–1267, 1999.

[7] T. Hoßfeld, K. Leibnitz, and A. Nakao, "Modeling of Modern Router Architectures Supporting Network Virtualization," in *2nd International Workshop on the Network of the Future (FutureNet II) in conjunction with IEEE GLOBECOM*, 2009.

[8] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Fairness issues in software virtual routers," in *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, 2008.

[9] IEEE, "IEEE802.3x Specification for 802.3 Full Duplex Operation," 1998.