ENDEAVOUR: D4.1 :Final Use Cases from related work

Christoph Dietzel, M. Bleidner, Thomas King, Mitch Gusat, Philippe Owezarski

ENDEAVOUR: Towards a flexible software-defined network ecosystem

ENDEAVOUR

Project name	ENDEAVOUR
Project ID	H2020-ICT-2014-1 Project No. 644960
Working Package Number	4
Deliverable Number	4.1
Document title	Final Use Cases from Related Work
Document version	1.0
Editor in Chief	Dietzel, DE-CIX
Authors	Dietzel, Bleidner, King, Gusat
Date	06/07/2015
Reviewer	UCLO
Date of Review	04/12/2015
Status	*Public*

Revision History

Date	Version	Description	Author
18/06/15	0.1	First draft	Dietzel, Bleidner
25/06/15	0.2	Integrated Sections 2.4 and 2.6.3.	Gusat, Kathareios
26/06/15	0.3	Improved structure, content and wording	Dietzel, Bleidner, King
19/06/15	0.4	Minor changes	Uhlig, Kathareios
02/07/15	0.5	Minor changes	Dietzel, Bleidner
03/07/15	0.6	Add SoA on unsupervised attack detection	Owezarski
26/11/15	0.7	Included list of acronyms	Dietzel, Bleidner
01/12/15	0.8	Added a summary	Dietzel, Bleidner
04/12/15	0.9	Review minor corrections	Canini, Chiesa
10/12/15	1.0	Final improved version	Dietzel, Bleidner

Executive Summary

By leveraging SDN technologies at the core of the inter-domain routing convergence point, namely IXPs, ENDEAVOUR addresses limitations of the network interconnection model in the current Internet. To engage the Internet stakeholders, ENDEAVOUR targets the rich ecosystem of one of the largest IXPs, DE-CIX, which we see as an ideal platform to implement new solutions and provide novel services based on the capabilities of SDN.

This comprehensive survey of use cases summarizes relevant work from related publications. We focus on collecting applicable use cases for deploying SDN at IXPs and additionally take into consideration work that one can envision to be beneficial at IXPs. The identified fields comprise traffic engineering, peering, security, new enabling services, monitoring applications, and new opportunities to optimize IXP management.

In contrast to today's peering environment, which is solely prefix-oriented, we envision peering for specific applications. By allowing forwarding rules that match any IP header field and thus traffic can be handled at the desired granularity. This also enables enhanced inbound traffic engineering schemes, which give a member more control over how the traffic enters his network. Moreover, we identified a multitude of security related use cases. Among other benefits, SDN-filtering mechanisms allow IXPs to be more robust against accidental and intentional misconfiguration. ENDEAVOUR strives to provide the technology for enabling new services at an IXP, by making use of SDN's separation of concerns. Thereby, we envision a higher rate of innovation at IXPs by developing new services such as centralized routing or traffic steering.

ENDEAVOUR will revisit the provided use cases in future deliverables to determine whether they are to be considered in the SDN architecture of ENDEAVOUR.

Contents

1 Introduction

The mission of ENDEAVOUR is to revolutionize the Internet eXchange Point (IXP) community with state-of-the-art, and beyond, Software Defined Networking (SDN) technologies.

The main hardware of the Internet, i.e., switches and routers, strongly couples the control plane with the data plane and bundles them on those devices. As a consequence, introducing innovation in the Internet protocol suite is slow and challenging, if not impossible in many cases. SDN is an emerging network paradigm to encourage innovation and overcome the previously mentioned limitations. SDN separates the control plane (making the forwarding decision) and the data plane (forwarding the actual traffic), previously integrated in a single networking device. This separation is realized by means of well-defined interfaces between the networking equipment and the logically centralized controller. OpenFlow [82] is the most prevalent implementation of the communication protocol (Southbound API) between the controller and the underlying hardware.

Interconnection of Internet Protocol (IP) networks relies on complex, static, and hard to manage mechanisms [16, 23]. The demand for more flexible and effective traffic management is increasing, due to the growing traffic volumes and the need for reaction times close to real-time. Over the last decade, IXPs became central elements of the Internet ecosystem. Given the fact that the largest IXPs carry similar amounts of traffic compared to the largest global ISPs [6], they ultimately came into focus of the Internet and network research community [25, 22, 97, 26]. So-called IXP members exchange Internet traffic with various other IXP members connected to the same IXP over a layer 2 switching fabric. The entity that operates the IXP infrastructure is referred to as IXP operator throughout this document.

Despite the still limited impact of SDN in practice [66], we believe that introducing SDN at IXPs is strategically sound and bears great potential. Even a single deployment of SDN technology at an IXP can have a large impact on a variety of stakeholders and their networks [118]. The large number of connected networks, which can benefit from novel and innovative services [109] implemented on the IXP, makes us confident that IXPs are the right place for implementing SDN based technology with the knowledge gathered through ENDEAVOUR. In this document we present use cases based on the novel techniques SDN can enable, supported by the state-of-the-art. Each use case offers potential benefits for different stakeholders.

The following section discusses the identified use cases and puts them into context.

2 Survey of Use Cases

In this section we provide a comprehensive survey of use cases for introducing SDN-based technology at an IXP. The use cases are extracted from state-of-the-art SDN applications proposed in a multiplicity of publications. We arranged those applications in six major categories, namely: traffic engineering (cf. Section 2.1), peering (2.2), safety and security (2.3), monitoring (2.4), management (2.5), and enabling services (2.6). The surveyed publications either describe SDN related use cases directly within the context of IXPs or more generally global inter-domain routing.

2.1 Traffic Engineering

The traditional goal of traffic engineering is to maximize aggregate network utilization, allow optimal load balancing, assure failover, and other traffic flow optimizations [10]. The rise of video streaming and cloud services has generated significantly higher traffic volumes, not only for the core ISPs in the Internet but also for edge networks [74]. This has led to fine-tuned routing policies, mostly implemented with the Border Gateway Protocol (BGP) [94]. While intra-domain routing protocols can be used inside an Autonomous System (AS) [46, 126], mechanisms based on BGP are widely used on the AS boundaries [84].

The SDN paradigm introduces two benefits that can be utilized for traffic engineering [58]. First, a logically centralized controller that has a global view of distributed network states [82]. Given this centralized view, the controller can optimally allocate network resources, including information about network resource limitations, dynamics in the network status as well as application requirements. Second, the programmability of networking hardware allows the dynamic transformation of the knowledge of the controller into rules on the forwarding devices.

Existing SDN deployments aim at improved wide-area traffic engineering, and the reported architecture results in better network capacity utilization and improved delay and loss performance [66]. Furthermore, we find several relevant SDN traffic engineering approaches in [44] and Akyildiz et al. present a roadmap for traffic engineering in SDN-OpenFlow networks [7]. However, SDN must not be considered as an all-or-nothing approach. Agarwal et al. demonstrated that an incremental and strategically deployed SDN environment can successfully co-exist with traditional networks and still improve the network performance significantly [5]. We believe that this holds true for the IXP environment as well. A deployed SDN environment can

be beneficial for the IXP operators as a first step, with the possibility of extending it to their members later.

2.1.1 Load Balancing

A common technique to balance traffic across clusters of servers is to manipulate the Domain Name System (DNS) frequently. A single DNS record resolves to several IP addresses for different backend systems. However, this approach comes with several limitations such as increasing latency and slower responses to failures [104]. Since the early days of the SDN paradigm it envisioned the application of load balancing. Therefore, different techniques have been proposed [59, 122, 27]. They focus on wildcard-based forwarding rules to be replaced for each flow by a meaningful IP address, and scalability to seamlessly add resources to a cluster.

Building on this, Gupta et al. consider in [58] the implementation of those approaches at IXPs to enable wide area server load balancing. Hence, an IXP member could announce a single anycast IP address so that any flow addressed to this IP address is redirected according to load-balancing policies depending on requested load and geographical location. The redirection includes packet header manipulations, which can be implemented based on SDN capabilities at an IXP.

2.1.2 Inbound Traffic Engineering

BGP is the prevalent inter-domain routing protocol in today's Internet. The routing decision is based on the destination IP address. Thus, ASes have limited control over how the traffic enters their networks. Mechanisms such as path prepending [93, 24], communities and selective announcements [94], originally not part of the BGP routing protocol, have been widely adopted to fill this gap. Additionally, ASes originating traffic may have their own policies in place for outbound traffic [115], limiting the ability to control traffic based on inbound traffic engineering [93]. At very dense traffic convergence points such as IXPs, a wide range of independent and inconsistent peering policies clash [99].

These limitations are an opportunity for inbound traffic engineering at IXPs. Gupta et al. propose in [58] allowing IXP members to install forwarding rules in SDN-enabled switches of the IXP. This enables ASes to directly control inbound traffic according to, for example, source IP addresses or port numbers. Generally, an SDN enabled IXP would allow installing flow rules that match any IP header field and thus inbound traffic can be handled on

the desired granularity. Existing SDN approaches (e.g., [111], [112]) can be integrated in an IXP SDN platform and allow ASes to deploy more efficient inbound traffic engineering policies.

2.2 Peering

Peering is the prevalent business relation between IXP members. On this basis the largest IXPs exchange up to 4 terabits per second of Internet traffic between hundreds of ASes [25]. These dense interconnection points are generally convenient locations to introduce new technologies.

While contemporary BGP peering allows fine-grained prefix-specific peering arrangements, more fine-grained peering can be envisioned as a highly effective advancement for high-bandwidth services, e.g., video streaming. A more concrete use case proposed by Gupta et al. in [58] suggests that an Internet Service Provider (ISP) could configure its edge routers to make different forwarding decisions for different application packet classifiers and routing policies. This supports a flexible inter-domain routing policy, and creates new business relationships between ASes, based on multiple match fields of IP headers not limited to the predominant destination prefixes anymore [124].

Considering that peering is the core business case for IXPs, we expect that more peering use cases will spring from the technical work within ENDEAVOUR.

2.3 Safety and Security

Networks are complex structures of individual interconnected infrastructure devices, which communicate based on distributed protocols. From a security perspective such a complex structure implies multiple possible attack vectors. Networks can be considered as desirable targets for malicious activities, since they can provide attackers with access to a large number of computer systems. Therefore, networks require carefully designed security mechanisms, such as access controls for proactively preventing attacks, but also monitoring functionality in order to uncover ongoing malicious activity within the network and update the access controls accordingly. A secure and safe operation must be ensured for both the IXP operator and IXP members. Safety is referred to as the protection from an unintended potentially harmful activity (e.g., misconfigured router). Security is referred to as the protection and detection of malicious activity (e.g., a Distributed Denial of Service (DDoS) attack).

Extensive research on network security by means of SDN has produced noticeable results [8, 102, 129, 15]. However, beside the potential security enhancements brought by SDN, it may also introduce new attack vectors, given the way networks are operated and managed [71, 106] with it. This section focuses on the use cases arising from a safety and security perspective in the context of IXP operators as much as IXP members.

From an operator point of view, there is a strong incentive to securely and safely operate its layer 2 switching fabric, which interconnects the individual members. In this context Section 2.3.3 covers the use cases for implementing filtering in order to drop unwanted traffic such as specific broadcast traffic.

From an IXP member perspective, use cases such as effective DDoS detection and mitigation or prevention of network capacity thefts [65] are relevant. We discuss related issues in Section 2.3.1 and Section 2.3.3.

2.3.1 Policy Support

Wide-area traffic exchange based on peerings [25] is mainly controlled by the capabilities provided by BGP, which was initially designed as a pure next-hop routing protocol, though widely used for traffic engineering purposes [94]. Due to increasing demands for policy support, BGP was extended to support policy functionalities [112]. From an IXP member perspective, the capabilities of controlling inter-domain routing based on BGP are limited to destination-based control. This way, a member can control outgoing traffic by influencing the BGP route selection process. However, a network has limited control over incoming traffic [94] (see Section 2.1).

Richer policy enforcements in order to support the novel peering relations described in Section 2.2 would be desirable. Members would be allowed to specify concrete policies, which are enforced on their behalf on ingress and egress ports at the IXP. The IXP would provide a central interface based on the SDN paradigm that offers support for specifying member related policies. Those policies would be gathered and checked for possible conflicts. After the checking process, a conflict free policy set would be installed within the infrastructure of the IXP. Efficiently installing such a policy set in a distributed network with multiple network devices, with vendor-specific configuration settings, is a challenging task. Leveraging the central control plane envisioned by the SDN paradigm in combination with a standardized interface provided by each network devices, such as OpenFlow, could simplify a network wide policy enforcement. Multiple approaches have already been proposed [119, 47] for network policy languages. These approaches describe a solution for specifying policies consistently throughout the network,

as well as evaluating them in terms of conflicting rule sets. Those policy languages also have the potential to simplify the management task for IXP operators, which is further discussed in Section 2.5.

Blackholing, which was recently introduced and implemented at various IXPs[1], is one example use case that can benefit from an extended policy support. Blackholing in this context describes the ability to discard unwanted traffic at the IXP's infrastructure triggered by the members, especially in case of ongoing DDoS attacks. For instance, current implementations leverage BGP for this. If a member experiences a high traffic volume for one of his announced prefixes, which is due to a possible DDoS attack, then the member can reactively set the BGP next hop for this prefix to be the IP address internally used by the IXP for blackholing. The required feature set is already implemented, but lacks support for more fine-grained control over which traffic should be directly discarded at the IXP's infrastructure. BGP currently limits this control at the granularity of full prefixes. Having the ability to specify a fine-grained subset of traffic based on, for instance, a source IP address would be desirable [98, 50]. Instead of relying on BGP, an IXP could provide its members with an SDN interface for specifying certain drop rules, to be directly installed within the IXP's infrastructure.

Beside the blackholing feature, extended policy support would also allow members to control their IXP facing in and outbound network interfaces in terms of allowed network paths or constraints on the port utilization rates [61].

2.3.2 Attack Detection

Section 2.3.1 introduces the use cases for being able to drop unwanted (potential attack) traffic directly at the IXP's infrastructure, in order to protect a member's own network infrastructure or port from congestion. Current approaches mostly support the mitigation of ongoing attacks but lack support of attack detection in two ways. First, it remains difficult to detect ongoing attacks. Second, it is difficult to determine when exactly a previously ongoing attack can be assumed to have been reduced in terms of volume. Given these two difficulties, it is challenging for an IXP member to estimate when exactly to trigger an attack mitigation feature, e.g. blackholing, and when to revoke such a feature, in order to minimize the inevitable loss of legitimate traffic.

Based on the current state-of-the-art, there is a use case for network operators to further improve attack mitigation with attack detection func-

[1] https://www.de-cix.net/products-services/de-cix-frankfurt/blackholing/

tionalities. Attacks targeting the availability of interconnected devices, such as DDoS attacks or crossfire attacks are still a serious threat to the Internet [38]. Detecting such attacks strongly relies on a detailed monitoring and classification of ongoing network traffic [128, 114]. Continuously monitoring and classifying network activities remains a challenging task, in particular with the growing size of many networks, growing amount of traffic traversing them [127], and the frequency of new attacks. This last item points out to one of the fundamental challenges in detecting network attacks, namely that attacks are a moving target. It is not possible to know the different attacks that an attacker may launch, because new attacks as well as new variants of already known attacks are continuously emerging. Indeed, attacks have become both increasingly numerous and sophisticated over the years. Classical static or supervised approach have severe limitations [79]. Motivated by the limitations of current knowledge-based approaches, a new research area has emerged in the last years, based on a diametrically opposite philosophy, for the detection of anomalous traffic events: Unsupervised Anomaly Detection. Instead of relying on a previously acquired knowledge on the characteristics of network attacks or on the baseline-traffic behavior, unsupervised detection uses data-mining techniques to extract patterns and uncover similar structures "hidden" in unlabeled traffic of unknown nature (attack or normal-operation traffic). Some methods for unsupervised detection of network attacks have been proposed in the past [91, 39, 79, 75]; the majority of them are based on clustering techniques and outliers detection.

With the rise of the SDN paradigm, attack detection based on more powerful monitoring (see Section 2.4) provided by SDN and OpenFlow in particular, can be envisioned. These functionalities are utilized in novel approaches in order to implement an enhanced anomaly detection and mitigation mechanism, in conjunction with SDN and well known monitoring tools such as sFlow [51]. Beside such more general approaches to counter anomalies within the network, other approaches focus on the detection of more specific attacks, such as DDoS flooding attacks [18] or crossfire attacks [52].

Enabling a more sophisticated attack detection supported by SDN features can be seen as a promising use case, mainly when deployed in the range of networks that includes the IXP infrastructure. This use case would be an important contribution to the currently available features within an IXP, such as blackholing, to mitigate the effects of DDoS attacks. Instead of purely relying on active triggering of such a mitigation feature by an IXP member, an attack detection mechanism within the IXP infrastructure can help IXP members in the efficient usage of such a mitigation feature, with

the potential of semi-automation of the process.

2.3.3 Filtering

Filtering unwanted traffic at a specific port is a use case inspired by the needs of IXPs and their layer 2 domain. In such a large layer 2 domain, broadcast traffic originated by ARP or IPv6 neighbor discovery protocols can significantly increase the workload on all connected routers [95, 88]. The lack of sufficient fine-grained filtering mechanisms also puts IXPs at risk of being affected by a misconfigured router operated by a member [65]. Therefore, IXPs usually publish configuration guidelines[2] that describe which traffic should not be forwarded to a port connected to the infrastructure of an IXP.

Due to their nature, guidelines do not necessarily prevent any misconfiguration. Therefore, the support for fine-grained filtering of traffic would be a desirable feature for an IXP. OpenFlow supports fine-grained filtering of traffic thanks to its flow-based forwarding scheme. Instead of relying on configuration guidelines, an IXP could specify which packets from members are allowed and which are directly dropped at the ingress port. It enables them to enforce their guidelines as policies and would increase the safety of the infrastructure. Already implemented approaches such as ARP sponge [17], which deals with exhaustive ARP requests within a large broadcasting domain, could be further enhanced by leveraging a central SDN controller. Packets, which might be intentionally sent by a connected router, but are not required to be forwarded to all other routers, can be forwarded to an SDN controller to be processed by the centralized control plane.

Besides the misconfiguration of individual routers, the switching platform is subject to additional security implications. Each connected member interconnects based on a BGP session with various other members, depending on their peering business model. Such a BGP session is established by the control plane, but is not necessarily enforced on the data plane [65]. For instance, a member is able to address and ultimately forward traffic to another member, regardless of the existence of a valid BGP session. It solely depends on the filtering settings of the receiving party, if this traffic is forwarded or dropped. If a member is frequently changing peering sessions with various other members, it can become a challenge to implement dynamic filtering rules on an access router. Having the ability to consistently implement the behavior determined by the control plane in terms of BGP

[2] https://www.de-cix.net/get-connected/technical-requirements/

sessions on the data plane can be a vital feature for IXP members to avoid unwanted traffic in their networks [11]. This way, only traffic flows that are made legitimate by a valid BGP session, would be allowed in an IXP network.

2.4 Monitoring

As with any SDN, the orchestration, management, load balancing, protection and isolation of the SDN-enabled IXP depends on timely access to the internal network state, including all the layers of the physical and virtual components. The area of network monitoring, telemetry and topography is exceedingly rich in literature. However, only a few schemes have been widely adopted and the current state-of-the-art in hardware network monitoring has remained limited to sampling a few, possibly isolated, links with a granularity in the 0.01s to 1s, implemented within approaches such as sFlow [121], NetFlow [30] and SNMP [21].

2.4.1 General Network Monitoring

Methodologically, IXP monitoring has been performed via SNMP, similarly to any other network. Monitoring the traffic that flows through the network is performed with the equally popularly-used capturing libraries such as Libcap[3]. For these, often additional hardware and special software configurations are required [117].

Another complementary form of global monitoring is per-path delay estimation, typically performed end-to-end in closed source-destination pairings. This could implicitly build into transports such as Transport Control Protocol (TCP) [64], or be performed explicitly to get a statistical estimation of the current RTT.

Approaches such as sFlow, IPFIX [31], Netflow and enhancements to NetFlow [40] facilitate control decisions on information gathered from monitoring the network using traffic matrix-based approaches.

2.4.2 SDN Monitoring

Lack of SDN observability has already been pinpointed in literature [14, 36]. Encapsulating traffic renders it opaque to network monitoring and security tools [14] since the capabilities of deep packet inspection are insufficient to cope with encapsulated traffic. Yet, an SDN-enabled IXP that performs

[3] http://www.tcpdump.org

Overlay Virtual Network (OVN) translation would be able to monitor traffic during the decapsulation/encapsulation process. Nevertheless, monitoring benefits from SDN capabilities, both on the switch (OpenFlow statistics, byte/packet counters, per-flow state) and on the edge of the network (OpenStack ceilometer for datacenter billing).

Proposed solutions for monitoring using SDN are abundant. Latency monitoring with OpenFlow [89] is a novel approach to monitoring per-link or even per-route latency leveraging the OpenFlow protocol messages. Similarly FlowSense [130] extracts information on a per-flow basis from the control traffic of OpenFlow packet_in and packet_out messages. OpenTM [113] emphasizes the importance of an accurate traffic matrix for capacity planning, traffic engineering and routing protocol configuration. It keeps track of all active flows within the network and evaluates the flow based counter provided by OpenFlow switches per path. But even OpenFlow can not solve all open questions regarding an efficient and comprehensive traffic matrix estimation [133]. Mahout et al. [37] present another OpenFlow-based monitoring mechanism to detect elephant flows and route the identified flows on the least congested links. Furthermore, OpenNetMon [117] monitors throughput, packet loss and delay between two endpoints using active measurements, leveraging the packet injection capabilities of the controller. Liteflow [56] distributes the load of monitoring flows among SDN switches, and makes the scalability and accuracy of network monitoring manageable.

The various approaches presented in recent publications leverage the unique capabilities of SDN to provide more accurate monitoring solutions, including a more detailed view of the current network state. Such a detailed view is vital for a number of services built on top of it, such as management and security services. Therefore, implementing an SDN based monitoring solution at the IXP is one basis requirement for further developing novel services.

2.5 Management

Network management is a complex and error prone task [125], given the fact that network management has to deal with low-level and vendor specific configurations of distributed interconnected networking devices. Those networking devices, such as router and switches, are often closed, proprietary, and vertically integrated [69]. Simplifying and easing the way networks are managed and configured is vital, since network management is a continuous process that requires adaptation to changing network conditions on demand. The use cases of applying SDN technology in this scope can be

divided into two areas. First, they concern enhancing and simplifying existing processes, where the performed management tasks can be automated. Second, they concern enabling novel features within the network that would neither be cost efficient nor even be possible given the current complexity of the network management process.

Simplifying existing processes of the daily management business includes operational and provisioning tasks, where complexity arises from the vendor-specific configuration and distributed nature of today's networking devices. SDN can contribute to simplifying this management process with its ability to centralize the network control and in addition can introduce an open and standardized configuration interface. A number of different proposals introduce network management languages [119, 62, 47, 107], which strive to simplify the network management by providing network operators a high-level policy language. The proposed high-level policy languages provide network operators with the ability to specify their network policies on a higher layer and especially independent from a specific network topology or vendor-specific configuration. The language itself is in charge of translating those specified policies into low-level configurations, which can be subsequently installed on the relevant network devices. Those policy languages have attracted increasing interest from the research community, since OpenFlow simplifies such a translation process. Swapping hand crafted configuration settings in favor of automatically generated configuration settings by a high-level policy language reduces the likelihood of misconfiguration within a network. Ultimately, it also enables extended automation of operational and provisioning tasks [68].

Proposals such as the Flow-based Management Language [62] have been used to manage multiple enterprises for over a year, proving their feasibility. Since operating an IXP infrastructure includes a number of management tasks, such as the provisioning of member ports or performing maintenance operations on the current infrastructure, there is a clear use case for IXPs to migrate their infrastructure towards an SDN, in order to benefit from SDN from a management point of view. Regarding the migration, there is often a lack of a clear understanding of how an existing infrastructure can be migrated towards an SDN infrastructure [44].

Approaches like Cardigan [109] envision a new layer of abstraction, where today's well-known routing protocols are executed in a virtual environment, which closely follows the underlying physical structure of a network. This way operators can keep their existing protocols within this virtual environment, combining the benefits of having a unified underlying physical infrastructure, with the well-known routing protocols. While also having

the ability to continuously innovate based on the functionality available at the SDN-enabled physical infrastructure.

In case of maintenance operations, it would be desirable for IXP operators to flexibly redirect traffic to avoid, for instance, one of its core switches. Having this ability would enable an IXP operator to more easily perform maintenance on various devices, without interfering with the traffic needs of members. Even though this use case is not directly addressed by the research community, the need for energy-aware routing for Datacenter (DC) operators [60, 70] is a similar use case, which could be applied to IXP operators as well.

2.6 Enabling Services

This section introduces three novel services, traffic steering, centralized routing and the extension of virtualized networks of clouds.

All three have potential to optimize traffic engineering, not only at IXPs, but in the entire Internet. The transition from the classical distributed networking and hardware centric approach to the more flexible and innovative SDN paradigm, could enable IXPs to extend their current service offer, which today is primarily enabling peering between various networks.

Having flexible control over the forwarding hardware can enable services which would not be possible with the current generation of hardware. This would be the case mainly because the management effort would be too large, which hinders the deployment of novel services in the relevant scope of IXPs. However, SDN has shown its potential to simplify network management and its ability to implement novel applications and services on top of the network [72].

2.6.1 Traffic Steering

Many network management tasks, such as firewalls, Deep Packet Inspection (DPI) or NATs rely on middleboxes. Such middleboxes are deployed somewhere in a network, e.g., at the edges of ISP networks [103, 58]. Each middlebox provides a unique service, which processes the traffic traversing this middlebox. Since some of these services (e.g. DPI) require extensive amounts of processing power, traffic should only be redirected through a middlebox if absolutely necessary, requiring a certain flexibility in the forwarding configuration. The process of forwarding a subset of the overall traffic within a network through a defined set of middleboxes is referred to as traffic steering. While middleboxes can provide enhanced network perfor-

mance and security benefits [103], various challenges in the deployment of such middleboxes are identified [92, 42]. One of those challenges is related to the location of a middlebox within a network [132]. Especially large ISPs, which operate a global network, have difficulties determining the right position within their network to place individual middleboxes, since the traffic related to a specific user group might be forwarded over a longer path in order to traverse the required middlebox.

Additionally, middleboxes are mostly statically deployed and configured, which makes it difficult for network operators to flexibly change certain parameters, such as the order of middleboxes, traversed by the traffic [92]. IXPs and especially their role as a central traffic exchange point could provide solutions for those challenges. Middleboxes could be deployed within the central IXP infrastructure, instead of deploying duplicate middleboxes at various locations within an ISP network [58].

Therefore, offering middlebox functionality at an IXP is identified to be a possible use case leveraging SDN capabilities [58]. This use case requires two key features: *flexible traffic forwarding* and *external control*. Both can be implemented by leveraging SDN technologies. The flexible forwarding of traffic is a key element of the SDN paradigm, which is enabled by the central control of SDN switches [132]. Traffic can be flexibly redirected through a certain chain of middleboxes. The second key capability is external control of this flexible forwarding behavior. External control should give a member connected to an IXP network the ability to control the internal forwarding behavior as far as necessary to redirect traffic through certain middleboxes deployed at the IXP. A possible interface for providing external control could be implemented using the extended policy support introduced in Section 2.3.1.

In the future traffic steering could be extended to a level where the middleboxes themselves can be implemented directly within the capabilities of SDN switches [78]. Having the ability to not only control the forwarding configuration, but also implement certain middleboxes on demand extends the possible use cases of traffic steering at an IXP, where an IXP might offer services to its customers on demand thanks to the flexibility and configurability enabled by SDN technologies.

2.6.2 Centralized Routing

Novel SDN implementations have been mostly focused on an intra-domain routing context, where routing is centralized within the boundaries of a single AS [70]. In this context, the central controller, which is responsible for

making routing decisions and propagating those decisions to each forwarding device within the network domain, is commonly owned by a single entity, which is responsible for the AS domain. Possible benefits of centralizing the routing decision process have already been envisioned even before the initial SDN paradigm [19, 43, 48, 123]. This early work also includes the vision to transfer those benefits to an inter-domain routing context. This strong interest is renewed with the SDN paradigm. The focus of separating the control plane from the data plane, opens up new possibilities for implementing a centralized routing scheme. The unique position of IXPs within the Internet ecosystem provides an opportunity for developing centralized routing approaches based on SDN principals.

As discussed in Section 2.1.2, BGP is today's widely used protocol for inter-domain routing, which was further extended over the recent years to cope with new challenges imposed by today's inter-domain routing requirements. There is an increasing number of BGP sessions and subsequently possible divergent paths, especially within an IXP peering LAN [25]. Therefore, the process of making optimal and efficient routing decisions becomes more and more complex for a distributed protocol such as BGP. Multiple works have been focused on the current challenges and limitations of BGP deployments, such as slow convergence of BGP in case of path failures [86, 73], which even increases due to mechanisms such as route flap damping [81] or route oscillation due to unstable prefixes [49].

The use case of providing a centralized routing scheme arises from those challenges implied by BGP and today's rich diversity of possible paths in the Internet, especially because of the increasing popularity of peering at an IXP [25]. Since the control plane is decoupled from the forwarding plane in SDN, routing decisions can be made on a centralized controller or even outsourced to a trusted third party [76, 70]. Following this approach, a centralized routing entity could make inter-domain routing decisions and subsequently calculate routing paths for different network owners who have authorized this entity to be responsible for their routing management. Having input from different networks (e.g., different ASes) increases visibility over different network topologies. Such an increased visibility results in more efficient routing decisions, in a way that optimal routes between two endpoints in the scope of the routing entity can be calculated. In addition, troubleshooting of routing problems should be easier [70]. Load balancing can be implemented taking path costs and performance metric into account [98]. In case of a link failure, the central controller is able to efficiently recalculate and propagate an alternative path, resulting in a faster failover recovery than would be the case with BGP [110]. In consequence such a

routing entity could help to evolve the current BGP-focused inter-domain routing to a next level, as well as simplifying the routing management itself [70].

Centralized routing in an inter-domain routing context is a challenging task, which requires further research into possible interaction models and data exchange between the involved parties. Therefore, widely deployed routing protocols such as BGP are nearly impossible to replace with completely new designed protocols or approaches.

However, Nascimento et al. propose in [85] to incrementally combine well-established routing protocols with upcoming SDN technologies to optimize inter-domain routing. The use case of centralized routing is an enabling technology to develop novel services such as virtualized IP routing [85, 76].

2.6.3 Extending the Virtual Networks of Clouds

Today's cloud is based on large aggregations of hardware (servers and storage) and software (hypervisors, operating systems and applications), distributed in scale out systems, i.e, DC [12, 27, 33, 53]. Typically, DCs are virtualized both on the server level (e.g Virtual Machine (VM)) and the network level (OVNs) [14, 27, 35], providing the capability to serve multiple tenants simultaneously and their users in isolation. An emerging trend in the literature is the need to build the Multi-cloud, an interconnection of multiple Performance Optimized Datacenters (PoDs) across IXPs [116, 29, 131, 27, 45, 87, 33, 14]. The cloud would then be based on the illusion of a single-roof, continental-scale (virtual) DC, which in reality is distributed in the interconnected PoDs. As each component PoD of the multi-cloud will have its own characteristics in terms of network virtualization and SDN, the IXP will have to extend the SDN/OVN of each PoD and/or provide translation among them. Furthermore, it will have to incorporate the key features of portability (migration from one PoD to another), interoperability, heterogeneity and geo-diversity [87].

In [28] a comprehensive survey of network virtualization in DCs is presented. It is apparent in the survey that the network virtualization techniques are abundant and their translation has to be performed at the IXP. Centrally extending them in the IXP provides benefits both for DCs and the IXP. Performing the translation at each DC would be inefficient, as the same translation resources would need to be present at each PoD. Instead, the PoDs could forward encapsulated traffic to the IXP which takes care of the translation to a different OVN, gaining two main benefits: the PoDs keep their interconnection interfaces as simple as possible and the IXP

becomes essential to the multi-cloud environment.

Since the survey was published, a number of novel network virtualization technologies and architectures have been proposed [80, 108, 20, 83]. SDN-based OVNs on the DC allow VM mobility and ease of management, and offer complete traffic isolation for improved security, rather than treating the virtual network as a dumb extension of the physical network [55].

Most of the proposed solutions focus on functional requirements for the network virtualization platform while delivering the traditional network management and configuration experience at the virtual level. For example, it is quite common for a virtual network to mimic or even to completely emulate an L2 broadcast domain or an L3 subnet, as is the case for Virtual Extensible LAN (VXLAN) [80], NVGRE [108] and Geneve [54]. As a result, the complexity and the fragility of traditional physical network configuration are often copied into the virtual environment. In addition, existing network virtualization solutions fail to answer the complete set of network virtualization platform requirements. For example, both VXLAN and NetLord [83] rely on data plane learning for location and address dissemination, inheriting well-known L2 flooding and stabilization upon change limitations. In [120], the impact of virtualization on network performance in datacenters is studied. As VXLAN is emerging as the de-facto standard for the future of SDN-based OVN/tunneling [4] for datacenter networks, its scalability problem is addressed by Ethernet Virtual Private Network (EVPN). EVPN is an emerging IETF standard [63, 100] that uses MP-BGP for MAC learning in the control plane, extending the VXLAN across a Wide Area Network (WAN) [4]. This trend raises the question of whether an SDN-enabled IXP could in the future enable multi-cloud services and platform-neutral *super*-overlays by extending these standards to interconnect two or more DCs.

At the forefront of the state-of-the-art in DC SDN overlays is zOVN [36], a zero-loss overlay virtual network that addresses the lossless assumption of converged multi-tenant DCs Converged Enhanced Ethernet [1, 3, 2]. For a multi-cloud IXP interconnected environment, the IXP is challenged with extending the zero-loss overlay among the DC components that require it.

2.6.4 Cloud Transports and Tunnels Optimization

Similarly to the need for extending the DC SDN/OVN in the IXP, there is a growing interest in extending the transport protocols used in DCs beyond the internal DC networks. Although the Internet is currently dominated by TCP and User Datagram Protocol (UDP), virtualized DCs move towards

performance-optimized transport protocols [67, 105, 12, 57, 90, 77, 13, 83, 32, 108, 80]. Datacenter TCP (DCTCP), Multi-path TCP (MPTCP) and Fast and Secure Protocol (FASP) are some prevalent examples.

DCTCP [9] is a TCP transport protocol developed by Microsoft for DC networks. DCTCP leverages Explicit Congestion Notification in the network to provide multi-bit feedback to the end hosts by estimating the fraction of marked packets. In doing so, DCTCP sources react to the extent of congestion, not just the presence of congestion as is the case in TCP. This finer level of control allows DCTCP to operate with very low buffer occupancies while simultaneously achieving high throughput. MPTCP [96, 101] is a modification of TCP that offers path redundancy by enabling the simultaneous use of several IP-addresses/interfaces. It presents a regular TCP interface to applications, while in fact spreading data across several subflows. MPTCP is ported to the linux kernel and, in existing DCs, it is ready to be deployed leveraging well-known technologies such as ECMP. In [96], the authors found that MPTCP outperforms TCP by a factor of three when there is path diversity. FASP [41] overcomes the performance bottleneck of TCP on movement of massive data especially for WANs with large bandwidth, high round-trip time and packet loss.

As leading edge in SDN-based tunneling transports and further building on zOVN [36], the authors of [35] take advantage of the lossless intra-datacenter network and propose zFabric, a slim hypervisor stack that increases the throughput of long flows and enforces fairness independent of the transport type, by deconstructing and reallocating the transport functionalities. zFabric combines the ubiquity of TCP with the performance of UDP and Remote Direct Memory Access, resulting in order-of-magnitude lighter protocol stacks, i.e., reduced energy consumption in the virtualized cloud, and faster flow completion times for big data workloads. The emergence of losslessness in DC fabrics as shown in these works leads to the challenge and the vexing question of whether it can be extended beyond a single DC/PoD through an SDN-enabled IXP. The main challenge is the need for tunnels with credit exchange across an IXP that extends the local PoD flow controls [3]. These capabilities are not available yet, except for some exploratory research [35, 36, 34].

3 Outlook

SDN creates the opportunity for solving issues arising from the complexity of managing and operating traditional networks. Additionally, the ability

to innovate in the networking area has become a challenging task. This also holds true for IXPs. They have grown to become key components of the Internet over the past decades. Their unique position within the Internet ecosystems has lead to a variety of interesting use cases, given their large number of interconnected networks and the large amount of traffic exchanged via these interconnections. Thus, we foresee great potential by applying the SDN paradigm to IXPs in the scope of ENDEAVOUR and beyond. This document provides a comprehensive survey of use cases obtained from related work. Additionally, we cover use cases from the literature, that can be envisioned but are not directly related to IXPs. This valuable input will be used to form ENDEAVOUR. Notably, future WP4 deliverables will benefit from these use cases and will feed its outcomes directly into the desired SDN architecture of WP2 and WP3.

4 Summary

This document provides an overview of use cases extracted from available literature. We classified these use cases into six main categories: traffic engineering, peering, safety and security, monitoring, management, and enabling services.

SDN concepts for improved and novel traffic engineering implementations are manifold within the literature. The ultimate goal is to optimize the aggregated network utilization. By leveraging the global view of an SDN controller, the traffic flow across the IXP network can be allocated in an optimized manner. Additionally, IXP members have more control on how they receive or send traffic over these ports, e.g., load balancing mechanisms.

Peering as the predominant business for IXPs also has the potential for enabling new use cases based on SDN technology. While BGP-based peering allows exchanging traffic based on the destination IP prefix, the fine-grained flow control mechanisms introduced by SDN could lead to novel peering relationships. The incentive for setting up peering sessions could be positively influenced by having finer control about what kind of traffic will be exchanged over peering links. For instance, such traffic could be limited to specific applications.

Leveraging the extended programmability of the network to establish enhanced mechanisms to secure the IXP network. Access control and filtering can enhance the security of shared layer 2 network resulting in a enhanced security for exchanging data between the connected members. Additionally, novel approaches to detect malicious traffic and DDoS attacks on the

network layer are developed based on the extended monitoring capabilities introduced by SDN and OpenFlow in particular.

The field of network monitoring, telemetry, and topography is exceedingly rich in literature. While SDN offers a variety of features to control traffic flows, it lacks observability. This has been addressed in several publications that focus on concepts and implementations for OpenFlow.

Simplifying existing processes of the daily management business of IXPs is a crucial aspiration of SDN. Existing work focuses on network management languages and introduces new layers of abstraction to ease configuration changes and deployment.

Finally, we introduce three novel services from related work; traffic steering, centralized routing, and the extension of virtualized networks of clouds.

5 Acronyms

SDN Software Defined Networking

BGP Border Gateway Protocol

ISP Internet Service Provider

IXP Internet eXchange Point

ISP Internet Service Provider

AS Autonomous System

IP Internet Protocol

DDoS Distributed Denial of Service

DNS Domain Name System

OVN Overlay Virtual Network

DPI Deep Packet Inspection

VM Virtual Machine

DCTCP Datacenter TCP

MPTCP Multi-path TCP

FASP Fast and Secure Protocol

TCP Transport Control Protocol

UDP User Datagram Protocol

EVPN Ethernet Virtual Private Network

DC Datacenter

PoD Performance Optimized Datacenter

VXLAN Virtual Extensible LAN

WAN Wide Area Network

References

[1] 802.1Qau - Virtual Bridged Local Area Networks - Amendment: Congestion Notification. Technical report, 2010.

[2] P802.1Qaz/D2.5 - Virtual Bridged Local Area Networks - Amendment: Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes. Draft standard, 2011.

[3] P802.1Qbb/D2.3 - Virtual Bridged Local Area Networks - Amendment: Priority-based Flow Control. Technical report, 2011.

[4] Learn About VXLAN in Virtualized Data Center Networks. http://www.juniper.net/techpubs/en_US/learn-about/LA_VXLANinDCs.pdf, 2015.

[5] S. Agarwal, M. Kodialam, and T. Lakshman. Traffic Engineering in Software Defined Networks. In *IEEE INFOCOM*, pages 2211–2219, 2013.

[6] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *ACM SIGCOMM Conference on Applications, technologies, architectures, and protocols for computer communication*, pages 163–174. ACM, 2012.

[7] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou. A Roadmap for Traffic Engineering in SDN-OpenFlow Networks. *Computer Networks*, 71:1–30, 2014.

[8] S. Ali, V. Sivaraman, A. Radford, and S. Jha. A Survey of Securing Networks Using Software Defined Networking. *IEEE Transactions on Reliability*, PP(99):1–12, 2015.

[9] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. DCTCP: Efficient Packet Transport for the Commoditized Data Center. In *ACM SIGCOMM Conference on Data Communication*, August 2010.

[10] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. Technical report, December 2001. Internet Draft.

[11] J. Bailey, D. Pemberton, A. Linton, C. Pelsser, and R. Bush. Enforcing RPKI-based routing policy on the data plane at an internet exchange. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 211–212. ACM, 2014.

[12] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. Towards Predictable Datacenter Networks. In *ACM SIGCOMM*, August 2011.

[13] H. Ballani, K. Jang, et al. Chatty Tenants and the Cloud Network Sharing Problem. In *Symposium on Networked Systems Design and Implementation (NSDI)*, April 2013.

[14] K. Barabash, R. Cohen, D. Hadas, V. Jain, R. Recio, and B. Rochwerger. A Case for Overlays in DCN Virtualization. In *Workshop on Data Center - Converged and Virtual Ethernet Switching (DC-CAVES)*, September 2011.

[15] A. Bates, K. Butler, A. Haeberlen, M. Sherr, and W. Zhou. Let SDN be your Eyes: Secure Forensics in Data Center Networks. In *Workshop on Security of Emerging Network Technologies (SENT)*, 2014.

[16] T. Benson, A. Akella, and D. Maltz. Unraveling the Complexity of Network Management. In *Symposium on Networked Systems Design and Implementation (NSDI)*, pages 335–348, 2009.

[17] V. Boteanu. Minimizing ARP traffic in the AMS-IX switching platform using OpenFlow. Master's thesis, 2013.

[18] R. Braga, E. Mota, and A. Passito. Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow. In *Local Computer Networks (LCN)*, pages 408–415. IEEE, 2010.

[19] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and Implementation of a Routing Control Platform. In *Symposium on Networked Systems Design & Implementation*, pages 15–28, 2005.

[20] M. Casado, T. Koponen, R. Ramanathan, and S. Shenker. Virtualizing the Network Forwarding Plane. In *Workshop on Programmable Routers for Extensible Services of Tomorrow*, page 8. ACM, 2010.

[21] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). RFC 1157, RFC Editor, 1990.

[22] I. Castro, J. Cardona, S. Gorinsky, and P. Francois. Remote Peering: More Peering without Internet Flattening. In *ACM International on Conference on emerging Networking Experiments and Technologies*, pages 185–198. ACM, 2014.

[23] I. Castro, A. Panda, B. Raghavan, S. Shenker, and S. Gorinsky. Route Bazaar: Automatic Interdomain Contract Negotiation. In *Workshop on Hot Topics in Operating Systems (HotOS)*, 2015.

[24] R. K. Chang and M. Lo. Inbound Traffic Engineering for Multihomed ASs using AS Path Prepending. *Network, IEEE*, 19(2):18–25, 2005.

[25] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. There is more to IXPs than meets the Eye. *ACM SIGCOMM Computer Communication Review*, 43(5):19–28, 2013.

[26] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. Quo vadis Open-IX? *ACM SIGCOMM Computer Communication Review*, 45(1):12–18, 2015.

[27] C. Chen, C. Liu, P. Liu, B. T. Loo, and L. Ding. A Scalable Multi-datacenter Layer-2 Network Architecture. In *ACM SIGCOMM Symposium on Software Defined Networking Research*, pages 8:1–8:12. ACM, 2015.

[28] N. M. K. Chowdhury and R. Boutaba. A Survey of Network Virtualization. *Computer Networks*, 54(5):862–876, 2010.

[29] K. Church, A. G. Greenberg, and J. R. Hamilton. On Delivering Embarrassingly Distributed Cloud Services. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 55–60. Citeseer, 2008.

[30] B. Claise. Cisco systems NetFlow Services Export Version 9. 2004.

[31] B. Claise. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. 2008.

[32] R. Cohen, K. Barabash, et al. An Intent-based Approach for Network Virtualization. In *IFIP/IEEE IM*, 2013.

[33] R. Cohen, K. Barabash, B. Rochwerger, L. Schour, D. Crisan, R. Birke, C. Minkenberg, M. Gusat, R. Recio, and V. Jain. An Intent-based Approach for Network Virtualization. In *Integrated Network Management (IM)*, pages 42–50. IEEE, 2013.

[34] D. Crisan, A. S. Anghel, R. Birke, C. Minkenberg, and M. Gusat. Short and Fat: TCP Performance in CEE Datacenter Networks. In *Symposium on High-Performance Interconnects (HOTI)*, August 2011.

[35] D. Crisan, R. Birke, N. Chrysos, C. Minkenberg, and M. Gusat. zFabric: How to Virtualize Lossless Ethernet? In *Cluster Computing (CLUSTER)*, pages 75–83. IEEE, 2014.

[36] D. Crisan, R. Birke, G. Cressier, C. Minkenberg, and M. Gusat. Got Loss? Get zOVN! In *ACM SIGCOMM*, August 2013.

[37] A. Curtis, W. Kim, and P. Yalagandula. Mahout: Low-overhead Datacenter Traffic Management using End-host-based Elephant Detection. In *IEEE INFOCOM*, pages 1629–1637, April 2011.

[38] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir. Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks. In *Internet Measurement Conference (IMC)*, pages 435–448. ACM, 2014.

[39] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. In *Apps. of Data Mining in Comp. Sec.*, 2002.

[40] C. Estan, K. Keys, D. Moore, and G. Varghese. Building a Better NetFlow. In *ACM SIGCOMM Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 245–256, 2004.

[41] X. Fan and M. Munson. Petabytes in Motion: Ultra High Speed Transport of Media Files A Theoretical Study and its Engineering Practice of Aspera fasp over 10Gbps WANs with Leading Storage Systems. In *SMPTE Conferences*, volume 2010, pages 2–13. Society of Motion Picture and Television Engineers, 2010.

[42] S. K. Fayazbakhsh, L. Chiang, V. Sekar, M. Yu, and J. C. Mogul. Enforcing Network-wide Policies in the Presence of Dynamic Middlebox Actions using FlowTags. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2014.

[43] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. Van Der Merwe. The case for Separating Routing from Routers. In *ACM*

SIGCOMM Workshop on Future Directions in Network Architecture, pages 5–12, 2004.

[44] N. Feamster, J. Rexford, and E. Zegura. The Road to SDN: An Intellectual History of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[45] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg. Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems. In *Cloud Computing (CLOUD)*, pages 887–894. IEEE Computer Society, 2013.

[46] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, October 2002.

[47] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker. Frenetic: A Network Programming Language. In *ACM SIGPLAN Notices*, volume 46, pages 279–291. ACM, 2011.

[48] J. Fu, P. Sjödin, and G. Karlsson. Intra-domain Routing Convergence with Centralized Control. *Computer Networks*, 53(18):2985–2996, 2009.

[49] V. Gill, D. McPherson, A. Retana, and D. Walton. Border Gateway Protocol (BGP) Persistent Route Oscillation Condition. 2002.

[50] K. Giotis, G. Androulidakis, and V. Maglaris. Leveraging SDN for Efficient Anomaly Detection and Mitigation on Legacy Networks. In *Software Defined Networks (EWSDN)*, pages 85–90. IEEE, 2014.

[51] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris. Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments. *Computer Networks*, 62:122–136, 2014.

[52] D. Gkounis, V. Kotronis, and X. Dimitropoulos. Towards Defeating the Crossfire Attack using SDN. *Computing Research Repository (CoRR)*, 2014.

[53] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The Cost of a Cloud: Research Problems in Data Center Networks. *ACM SIGCOMM Computer Communication Review*, 39(1):68–73, 2008.

[54] J. Gross, T. Sridhar, P. Garg, C. Wright, and I. Ganga. Geneve: Generic Network Virtualization Encapsulation. Internet Draft, 2014.

[55] H. Grover, D. Rao, D. Farinacci, and V. Moreno. Overlay Transport Virtualization. *Internet Engineering Task Force, Internet Draft*, 2011.

[56] N. Grover, N. Agarwal, and K. Kataoka. LiteFlow: Lightweight and Distributed Flow Monitoring Platform for SDN. In *Network Softwarization (NetSoft)*, pages 1–9, April 2015.

[57] C. Guo, G. Lu, et al. SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. In *COnference on emerging Networking EXperiments and Technologies (CoNEXT)*, November 2010.

[58] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, pages 551–562, 2014.

[59] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari. Plug-n-Serve: Load-balancing Web Traffic using OpenFlow. *ACM SIGCOMM Demo*, 4(5):6, 2009.

[60] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving Energy in Data Center Networks. In *Symposium on Networked Systems Design and Implementation (NSDI)*, volume 10, pages 249–264, 2010.

[61] T. Hinrichs, N. Gude, M. Casado, J. Mitchell, and S. Shenker. Expressing and Enforcing Flow-Based Network Security Policies. Technical report, 2008.

[62] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker. Practical Declarative Network Management. In *Workshop on Research on Enterprise Networking*, pages 1–10. ACM, 2009.

[63] A. Isaac, N. Bitar, J. Uttaro, R. Aggarwal, and A. Sajassi. BGP MPLS-Based Ethernet VPN. Technical report, 2015.

[64] V. Jacobson. Congestion Avoidance and Control. In *Symposium Proceedings on Communications Architectures and Protocols*, pages 314–329, 1988.

[65] M. Jager. Securing IXP Connectivity, 2012.

[66] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al. B4: Experience with a Globally-Deployed Software Defined WAN. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 3–14. ACM, 2013.

[67] V. Jeyakumar, M. Alizadeh, D. Mazieres, B. Prabhakar, C. Kim, and A. Greenberg. EyeQ: Practical Network Performance Isolation at the Edge. In *Symposium on Networked Systems Design and Implementation (NSDI)*, April 2013.

[68] H. Kim, T. Benson, A. Akella, and N. Feamster. The Evolution of Network Configuration: A Tale of Two Campuses. In *ACM SIGCOMM Conference on Internet Measurement Conference (IMC)*, pages 499–514, 2011.

[69] H. Kim and N. Feamster. Improving Network Management with Software Defined Networking. *Communications Magazine, IEEE*, 51(2):114–119, 2013.

[70] V. Kotronis, X. Dimitropoulos, and B. Ager. Outsourcing the routing control logic: Better internet routing based on sdn principles. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 55–60, 2012.

[71] D. Kreutz, F. Ramos, and P. Verissimo. Towards Secure and Dependable Software-Defined Networks. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 55–60, 2013.

[72] 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*, 103(1):14–76, 2015.

[73] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. *ACM SIGCOMM Computer Communication Review*, 30(4):175–187, 2000.

[74] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-domain Traffic. In *ACM SIGCOMM*, pages 75–86. ACM, 2010.

[75] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies Using Traffic Feature Distributions. In *ACM SIGCOMM*, 2005.

[76] K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford. *Routing as a Service*. Computer Science Division, University of California Berkeley, 2004.

[77] V. T. Lam, S. Radhakrishnan, R. Pan, A. Vahdat, and G. Varghese. NetShare and Stochastic NetShare: Predictable Bandwidth Allocation for Data Centers. *ACM SIGCOMM Computer Communication Review*, 42(3):6–11, July 2012.

[78] J. Lee, J. Tourrilhes, P. Sharma, and S. Banerjee. No more Middlebox: Integrate Processing into Network. *ACM SIGCOMM Computer Communication Review*, 41(4):459–460, 2011.

[79] K. Leung and C. Leckie. Unsupervised Anomaly Detection in Network Intrusion Detection Using Clustering, 2005.

[80] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright. VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Internet Draft, Internet Engineering Task Force, August 2011.

[81] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 221–233. ACM, 2002.

[82] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[83] J. Mudigonda, P. Yalagandula, J. C. Mogul, B. Stiekes, and Y. Pouffary. NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters. In *ACM SIGCOMM Conference on Data Communication*, August 2011.

[84] W. Mühlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel. In Search for an Appropriate Granularity to Model Routing Policies. In *ACM SIGCOMM*, 2007.

[85] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. Corrêa, S. C. de Lucena, and M. F. Magalh aes. Virtual Routers as a Service: The Routeflow Approach leveraging Software-Defined Networks. In *Conference on Future Internet Technologies*, pages 34–37. ACM, 2011.

[86] R. Oliveira, B. Zhang, D. Pei, and L. Zhang. Quantifying Path Exploration in the Internet. *IEEE/ACM Transactions on Networking*, 17(2):445–458, 2009.

[87] F. Paraiso, N. Haderer, P. Merle, R. Rouvoy, and L. Seinturier. A Federated Multi-cloud PaaS Infrastructure. In *Cloud Computing (CLOUD)*, pages 392–399, June 2012.

[88] I. Pepelnjak. Could IXPs Use OpenFlow To Scale?, 2013.

[89] K. Phemius and M. Bouet. Monitoring Latency with OpenFlow. In *Network and Service Management (CNSM)*, pages 122–125, 2013.

[90] L. Popa, G. Kumar, M. Chowdhury, et al. FairCloud: Sharing the Network in Cloud Computing. In *ACM SIGCOMM*, 2012.

[91] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion Detection with Unlabeled Data Using Clustering. In *ACM DMSA Workshop*, 2001.

[92] P. Quinn and T. Nadeau. Problem Statement for Service Function Chaining. RFC 7498, RFC Editor, 2015.

[93] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig. A Performance Evaluation of BGP-based Traffic Engineering. *International Journal of Network Management*, 15(3):177–191, 2005.

[94] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig. Interdomain Traffic Engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128, 2003.

[95] J. Rabadan, S. Sathappan, K. Nagaraj, W. Henderickx, T. King, and D. Melzer. Proxy-ARP/ND Function in EVPN Networks. Internet Draft, 2015.

[96] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving Datacenter Performance and Robustness with Multipath TCP. *ACM SIGCOMM Computer Communication Review*, 41(4):266–277, 2011.

[97] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger. Peering at Peerings: On the Role of IXP Route Servers. In *Internet Measurement Conference*, pages 31–44. ACM, 2014.

[98] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Corrêa, S. Cunha de Lucena, and R. Raszuk. Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 13–18. ACM, 2012.

[99] P. S. Ryan and J. Gerson. A primer on internet exchange points for policymakers and non-engineers. 2012.

[100] A. Sajassi, R. Aggarwal, J. Uttaro, N. Bitar, W. Henderickx, and A. Isaac. Requirements for Ethernet VPN (EVPN). Technical report, 2014.

[101] M. Scharf and A. Ford. Multipath TCP (MPTCP) Application Interface Considerations. Internet Draft, 2013.

[102] S. Scott-Hayward, G. O'Callaghan, and S. Sezer. SDN Security: A Survey. In *Future Networks and Services (SDN4FNS)*, pages 1–7, Nov 2013.

[103] V. Sekar, S. Ratnasamy, M. K. Reiter, N. Egi, and G. Shi. The Middlebox Manifesto: Enabling Innovation in Middlebox Deployment. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, page 21. ACM, 2011.

[104] A. Shaikh, R. Tewari, and M. Agrawal. On the Effectiveness of DNS-based Server Selection. In *IEEE INFOCOM*, volume 3, pages 1801–1810. IEEE, 2001.

[105] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha. Sharing the Data Center Network. In *Symposium on Networked Systems Design and Implementation (NSDI)*, April 2011.

[106] S. Shin and G. Gu. Attacking Software-Defined Networks: A First Feasibility Study. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 165–166. ACM, 2013.

[107] R. Soulé, S. Basu, R. Kleinberg, E. G. Sirer, and N. Foster. Managing the Network with Merlin. In *ACM SIGCOMM Workshop on Hot*

Topics in Software Defined Networking (HotSDN), pages 24:1–24:7. ACM, 2013.

[108] M. Sridharan, K. Duda, I. Ganga, A. Greenberg, G. Lin, M. Pearson, P. Thaler, C. Tumuluri, N. Venkataramaiah, and Y. Wang. NVGRE: Network Virtualization using Generic Routing Encapsulation. Internet Draft, Internet Engineering Task Force, September 2011.

[109] J. Stringer, D. Pemberton, Q. Fu, C. Lorier, R. Nelson, J. Bailey, C. N. Correa, C. Esteve Rothenberg, et al. Cardigan: SDN Distributed Routing Fabric going Live at an Internet Exchange. In *Computers and Communication (ISCC)*, pages 1–7. IEEE, 2014.

[110] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford. Network Architecture for Joint Failure Recovery and Traffic Engineering. *ACM SIGMETRICS Performance Evaluation Review*, 39(1):97–108, 2011.

[111] P. Sun, L. Vanbever, and J. Rexford. Scalable Programmable Inbound Traffic Engineering. *ACM SIGCOMM*, 2015.

[112] P. Thai and J. C. de Oliveira. Decoupling Policy from Routing with Software Defined Interdomain Management: Interdomain Routing for SDN-based Networks. In *Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, 2013.

[113] A. Tootoonchian, M. Ghobadi, and Y. Ganjali. OpenTM: Traffic Matrix Estimator for OpenFlow Networks. In *Passive and Active Measurement*, pages 201–210. Springer, 2010.

[114] H. Tsunoda and G. M. Keeni. Security by Simple Network Traffic Monitoring. In *Conference on Security of Information and Networks*, pages 201–204. ACM, 2012.

[115] S. Uhlig and O. Bonaventure. Designing BGP-based Outbound Traffic Engineering Techniques for Stub ASes. 34(5):89–106, October 2004. ACM Computer Communication Review.

[116] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez. Greening the Internet with Nano Data Centers. In *Conference on Emerging Networking Experiments and Technologies*, pages 37–48. ACM, 2009.

[117] N. L. Van Adrichem, C. Doerr, F. Kuipers, et al. Opennetmon: Network Monitoring in Openflow Software-Defined Networks. In *Network Operations and Management Symposium (NOMS)*, pages 1–8. IEEE, 2014.

[118] L. Vanbever. Novel Applications for a SDN-enabled Internet Exchange, 2013.

[119] A. Voellmy, H. Kim, and N. Feamster. Procera: A Language for High-Level Reactive Network Control. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, pages 43–48. ACM, 2012.

[120] G. Wang and T. S. E. Ng. The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. In *Conference on Computer Communications (IEEE INFOCOM)*, March 2010.

[121] M. Wang, B. Li, and Z. Li. sFlow: Towards Resource-Efficient and Agile Service Federation in Service Overlay Networks. In *Distributed Computing Systems*, pages 628–635. IEEE, 2004.

[122] R. Wang, D. Butnariu, J. Rexford, et al. OpenFlow-based Server Load Balancing Gone Wild, 2011.

[123] Y. Wang, I. Avramopoulos, and J. Rexford. Design for Configurability: Rethinking Interdomain Routing Policies from the Ground up. *Selected Areas in Communications*, 27(3):336–348, 2009.

[124] Y. Wang, J. Bi, and P. Lin. SRP: Building a Software Defined Interdomain Routing Plane via SDN.

[125] Y. Wang and I. Matta. Sdn management layer: Design requirements and future direction. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 555–562. IEEE, 2014.

[126] Y. Wang, Z. Wang, and L. Zhang. Internet Traffic Engineering without Full Mesh Overlaying. In *IEEE IEEE INFOCOM*, April 2001.

[127] K. Xu, F. Wang, and H. Wang. Lightweight and Informative Traffic Metrics for Data Center Monitoring. *Journal of Network and Systems Management*, 20(2):226–243, June 2012.

[128] K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Internet Traffic Behavior Profiling for Network Security Monitoring. *IEEE/ACM Transactions on Networking*, 16(6):1241–1252, Dec. 2008.

[129] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang. Enabling Security Functions with SDN: A Feasibility Study. *Computer Networks*, 2015.

[130] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha. Flowsense: Monitoring Network Utilization with Zero Measurement Cost. In *Passive and Active Measurement*, pages 31–41. Springer, 2013.

[131] Q. Zhang, L. Cheng, and R. Boutaba. Cloud Computing: state-of-the-art and Research Challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

[132] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patneyt, M. Shirazipour, R. Subrahmaniam, C. Truchan, and M. Tatipamula. Steering: A software-defined networking for inline service chaining. In *IEEE Network Protocols (ICNP)*, pages 1–10, Oct 2013.

[133] Q. Zhao, Z. Ge, J. Wang, and J. Xu. Robust Traffic Matrix Estimation with Imperfect Information: Making Use of Multiple Data Sources. *ACM SIGMETRICS*, 34(1):133–144, 2006.