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	4.2
Document title	Design of Use Cases for Operators of IXPs
Document version	1.0
Editor in Chief	Bleidner, DE-CIX
Authors	Bleidner, Dietzel
Date	28/01/2016
Reviewer	Chiesa, UCLO
Date of Review	21/01/2016
Status	*Public*

Revision History

Date	Version	Description	Author
03/12/15	0.1	Initial version	Bleidner, Dietzel
15/12/15	0.2	Added content to use cases	Bleidner, Dietzel
23/12/15	0.3	First Draft	Bleidner, Dietzel
06/01/16	0.4	Revised structure	Bleidner, Dietzel
12/01/16	0.5	Minor changes	King, Bleidner
14/01/16	0.6	Added figure to 3.2	Bruyere
15/01/16	0.7	Finalizing content	Bleidner, Dietzel
18/01/16	0.8	Added summary and outlook	Bleidner, Dietzel
21/01/16	0.9	Review	Chiesa
25/01/16	1.0	Implemented review feedback	Bleidner

Executive Summary

IXPs are convergence points for inter-domain routing, making them an integral part of the rich Internet ecosystem. They are interconnecting a multitude of different network types and easing the setup of peering relations. ENDEAVOUR strives to transform this ecosystem with innovative and disruptive ideas through the introduction of the SDN technology at IXPs.

In this Deliverable, we describe the set of use cases that address the current limitations of IXP networks. Furthermore, we present solutions based on the programmability and flexibility that SDN brings to the networking space. Based on the insights that we gained by operating a large scales IXP like DE-CIX, we identify three areas where SDN can have significant impact on transforming operational tasks: Safety & Security, IXP Management and Infrastructure.

We introduce a set of use cases that aim at increasing the reliability of IXP networks on the network layer, by accessing layer 3 header information through SDN technology. Thus, further increasing the sound and safe operation of large scale networks.

Moreover, we believe that SDN will play a key role in unifying the configuration interfaces of networking devices. Ultimately, this will allow IXP operators to implement a central configuration and management instance even across a multi-vendor infrastructure.

IXPs' networks carry a huge amount of peering traffic with peaks of up to five Tbps. Thus, for network design it is crucial to scale with further growth. We introduce SDN concepts for extending load balancing mechanisms well-known to IXP operators today. To cope with the enormous traffic growth, sharing the load over multiple paths becomes increasingly important. Likewise, we anticipate that the protocol stack in large IXP networks can be simplified with an SDN-like layer 2 label switching design, facilitating a reduced operational complexity.

ENDEAVOUR will evaluate the potential impact of the use cases provided in this deliverable. Based on this, we will make a selection of the most promising use cases to be considered for being implemented on top of the ENDEAVOUR architecture. Thus, a selection of use cases from this deliverable, as well as from Deliverable 4.3, will be implemented for demonstration purposes. This will allow ENDEAVOUR to show the practical impacts and relevance of SDN for both IXP operators and IXP members.

Contents

List of Tables

List of Figures

1 Introduction

While the Internet continues to evolve, today's applications require increasingly higher demands in bandwidth, lower latency, and higher availability. Driven by such requirements, two interesting aspects of the Internet ecosystem came into the focus of the research community in the past years, i.e., Software Defined Networking (SDN) and Internet eXchange Points (IXPs).

SDN is emphasized as the final breakthrough for more programmable computer networks. To offer higher programmability, the control plane and the data plane are separated. A logical centric entity controls multiple data plane devices inside a network. The OpenFlow protocol [18] is the most prevalent implementation of this concept. However, the practical impact falls behind the opportunities envisioned by academia. Most deployments of SDN technology occur in closed and controlled environments, e.g., data centers [29] or intra-domain routing [16]. We believe that SDN will enable network innovation and deployments beyond closed systems. Indeed, we believe that dense inter-domain routing hotspots can benefit from SDN.

Presently, hundreds of IXPs allow thousands of ASes to peer with each other [11]. The largest among them carry about five Tbps and count over 600 member networks with a sustainable growth for the next years. Most IXPs operate route servers [26] to foster as much open peering relations as possible. However, BGP-based routing solely focuses on reachability and allows only a very myopic view of the data plane [4]. This constrains the ability of networks to route their traffic in a more effective manner and limits innovation potential for novel services.

Combining SDN as a powerful new technology with the rich inter-domain routing ecosystem at IXPs culminates in a hotbed of innovation. First, enhanced programmability even at a single IXP enables up to hundreds of Autonomous Systems (ASes) to innovate their peering strategies. Second, deploying SDN at IXPs is strategically sound because the network setups of IXPs itself are quite static and scale with current SDN-capable switches.

ENDEAVOUR strives to impact the peering ecosystem at large by bringing SDN with practical use cases to IXPs. Fueled by numerous discussions, with input from workshops, a podium discussion and related work we present where exactly we expect SDN at IXPs to be beneficial. This Deliverable reflects the current state of ENDEAVOUR use cases and its potential benefits for IXP operators.

Section	Use Case Name	Category	Page
3.1	Access Control	Safety & Security	6
3.2	Network Resource Security	Safety & Security	8
3.3	Broadcast Prevention	Safety & Security	11
4.1	Central Configuration	IXP Management	12
4.2	Adaptive Monitoring	IXP Management	15
5.1	Load Balancing	IXP Infrastructure	17
5.2	Layer 2 Label Switching	IXP Infrastructure	21

Table 1: Overview of use cases for IXP operator.

2 Outline

In this section we briefly introduce the structure of this Deliverable. To ease the reading of this document, each use cases for IXP operators is structured with three paragraphs: *i*) we provide an overview of the problem, the current situation, and discuss its limitations., *ii*) we highlight the already available solutions and explain how they fail to address IXP operators everyday challenges, and *iii*) we aim to sketch an SDN solution to the problem, describe its technical implementation, and provide a brief description of the SDN features we want to take advantage of.

We present a comprehensive list of all use cases in Table 1. It lists the section in this document where the use case can be found, the name, its category, and on which page the description starts.

3 Safety & Security

The following section describes SDN use cases related to the safety and security categories. We describe how these use cases increase the operational safety (e.g., prevent unintended misconfiguration) as well as the overall security (e.g., secure route server access against attacks) of an IXP network.

3.1 Access Control

Current Situation

To ensure a secure and safe operation of an IXP network, which interconnects multiple hundreds of networks, an IXP operator has to carefully control the platform. This includes monitoring and enforcing who is allowed to send which kind of traffic via his network. We identified drawbacks of

currently deployed access control lists. SDN has the potential to increase the level of security and safety. It allows to further limit the allowed traffic exchanged via an IXP network, while it filters packets due to misconfiguration of a member's router.

IXPs maintain a shared layer 2 switching-fabric, where each member connects its router. In principle each member can exchange all kinds of Ethernet frames with any other member. However, the IXP operator usually only permits certain kinds of Ethernet frames [1]. Each ingress port of an IXP network has a certain Access Control List (ACL) assigned to limit the allowed Ethernet frames to e.g. 0x0800 IPv4.

With today's hardware deployed at DE-CIX ACLs are limited to restrict the EtherType and source Message Authentication Code (MAC) addresses. Filtering of other packets, such as OSPF, STP or other layer 4 management protocols is usually not possible.

In addition, each member is assigned with a unique Internet Protocol (IP) address from an IP range associated with the IXP. Enforcing the member's router to only use this assigned IP address when originating control plane packets (e.g. for communicating with the route server) is also challenging, since layer 3 information cannot be evaluated during a layer 2 ACL matching.

Available Solutions

Today's hardware, including the major vendors, usually limits the expression of ACL to the interface type. ACLs assigned to a layer 2 interface within an IXP context are limited to layer 2 information. Layer 3 and above are only available in configured layer 3 interfaces. However, some vendors (e.g., Alcatel-Lucent) announced that upcoming software releases will be able to access information from layer 2 and 3 in a single ACL.

Technical Description

Current networking hardware implements an allow-by-default scheme. Hence, by default, a packet is forwarded if it is not blocked by an ACL rule. The SDN paradigm and in particular the flow-based forwarding scheme of OpenFlow is different. OpenFlow implements a deny-by-default scheme. Thus, a packet is only forwarded if it matches a specific flow rule. Otherwise it will be dropped by default. The latest OpenFlow standard 1.5.1 [20] specifies 44 match fields, enabling a flow to match the packet header fields from layer 2 up to layer 4. It is worth noting that only 12 match fields

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

are required by the OpenFlow standard to be implemented by a vendor. However, available OpenFlow hardware sometimes also offers support for some of the optional matching fields.

Furthermore, OpenFlow defines a drop action. If packets match to a coarse grained flow rule, more specific flow rules with a associated drop action can drop a subset of packets which would match this coarse grained flow rules. With the combination of forwarding and drop actions, we can implement a combination of while- and black-listing forwarding. This enables a more expressive access control filtering. We can craft flow rules to only forward allowed packets according to the requirements specified by an IXP operator. Packets sent via a member router due to misconfiguration or malicious intent, which does not comply with the requirements of the IXP, can directly be filtered at the ingress port of the IXP network by means of L2-L4 forwarding rules.

Nevertheless, it remains challenging to filter specific management protocols such as OSPF solely with access to header information. Since some of these management protocols operate on layer 4 without well-defined port numbers, it is difficult to match those packets.

3.2 Broadcast Prevention

Current Situation

Network devices connected within a layer 2 network heavily rely on broadcast messages to keep their mapping between IP and MAC addresses up-to-date. However, while scaling a network (i.e., a single Ethernet broadcast domain) to hundreds or thousands of connected devices, broadcasting messages becomes an issue [3, 9]. The figure 1 depicts the level of broadcast ARP/Neighbor Discovery (ND) packet rate seen at AMS-IX in a period of 15 month, which keeps increasing even with current ARP mitigation techniques (e.g., ARP Sponge).

With a steadily growing number of connected member routers, the number of broadcasting packets inside the network increases for two reasons: first, each newly connected member router issues its own broadcasting packets. Second, each broadcast packets is duplicated for each connected member. With DE-CIX having currently more than 600 connected member routers in its layer 2 IXP network, the number of ARP/ND packets becomes a burden for the routers. Especially because ARP/ND packet handling requires non-negligible router's CPU utilization. Since those CPUs only have very limited processing power, a large number of ARP/ND can already exhaust their capabilities [13].

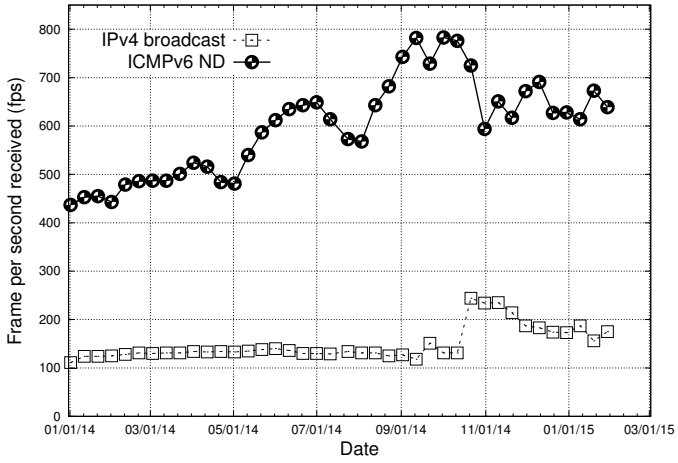

Figure 1: ARP IPv4 and ICMPv6 packet per second rate for a period of 15 months at AMS-IX.

The design context of ARP/ND does not apply for IXP networks altogether. The mapping between an interface IP address and its associated MAC address are known to the IXP operator. Furthermore, the MAC to IP mapping is rather static and only changes in case a new member connects to the IXP, a member replaces his router, or a member disconnects from an IXP. However, each member router still relies on ARP/ND to maintain the mapping between IP and MAC addresses, since there is no alternative available.

Available Solutions

The issue of growing broadcast traffic within an IXP network have already been addressed within multiple concepts [24, 3]. Some of them are applied in production networks, whereas the SDN based concepts have not been deployed yet.

ARP sponges: IXPs have developed partial solutions such as ARP sponges, which cannot directly prevent or reduce broadcast traffic. Instead, an ARP sponge is aware of all MAC to IP mappings and replies to ARP queries for unknown MAC or IP addresses within the network. Thereby, an ARP sponge prevents the circulation of ARP packets, which are not answered by any member router. However, the ARP/ND packets for known IP addresses within the network are unaffected and their quantity can still be an issue.

Proxy ARP/ND: A recent Internet Draft [24], describes the concept of using Ethernet Virtual Private Network (EVPN) [28] capabilities to tackle the exploding number of broadcasting packets in large layer 2 networks. The known MAC to IP address mappings can be distributed to all edge switches of an IXP network. ARP/ND requests, which arrive first at the edge switches, can be replied to on behalf of the actual address owner leveraging the available mapping information. This prevents those requests from being broadcasted through the IXP network.

Implementing a proxy ARP behavior with EVPN capabilities is a promising solutions for IXPs. However, the concept is still in the design stage missing the implementation of network hardware vendors.

Centralized ARP/ND Handling: ARP sponges usually lack the ability to efficiently reduce ARP/ND broadcast traffic. SDN and in particular OpenFlow offers the ability to control the forwarding behavior of individual packets. Given this fine-grained forwarding control, concepts have been discussed to implement centralized ARP/ND handling using OpenFlow [3]. ARP/ND requests can be redirected to a central instance, which has access to a global MAC to IP address mapping. Such an instance can reply to each request on behalf of the actual address owner, eliminating the need for broadcast traffic.

As stated in [3] deploying such an OpenFlow based approach is not a matter of available software but rather depending on the available hardware in production networks. Without OpenFlow capable hardware installed, it is not feasible. Nevertheless, such a solution is considered to be a perfect approach for ENDEAVOUR to built upon.

Technical Description

Given the flow-based forwarding scheme of OpenFlow, we introduce another possible solution in addition to the central handling.

Unicast towards the requested router: The first approach relies on a central instance to answer ARP/ND requests on behalf. Instead another approach transforms broadcasting traffic into unicast traffic towards the destination router which holds the requested address. As a result, an ARP/ND request is not destined to all connected member routers, but only to the one assigned with the requested IP address. Therefore, ARP/ND has to be detected at the ingress switch including the requested MAC address, which is supported since OpenFlow 1.3. The appropriate flow rule can match on

a specific requested IP address within the ARP/ND packet and exclusively forward the packet to the owner of the requested IP address. Since this information is known to the IXP operator, it can be proactively stored in flow rules inside the switching fabric, eliminating the delay imposed by reactive flow installation.

3.3 Network Resource Security

Current Situation

IXPs developed value-added services that require certain resources to be hosted within the IXP network (e.g., the route server). Since these resources are indispensable for a continuous operation of the IXP, security measures should be implemented directly at the network level.

The IXP network consists of all connected member routers and additionally certain resources hosted by the IXP within the same layer 2 domain. These resources (e.g., route server, provisioning hosts, monitoring systems) are required for a fully operational IXP. Route servers are a good example for mission critical resources [26], where each member receives BGP routing information from a centralized entity. Because of its importance for the IXP business, the Route Server must be secured against attacks and misuse. Implementing effective security measures within such a shared networking domain on a network level is a complex and challenging task.

Available Solutions

Even though resources hosted within the IXP network are only reachable by the connected members, they remain a potential attack surface for sabotaging an IXP's operation. Occasionally, they are even reachable from the outside the IXP network, due to route leaks from individual members. Limiting the rate of traffic forwarded to a resource would be one possible solution to mitigate Denial of Service (DoS) attacks on a certain resource. Current hardware can in principle implement rate limitings, however it often lacks support for rate limiting based on IP addresses in case the interfaces are configured in layer 2 mode (cf. Section 3.1).

Technical Description

OpenFlow offers both fine-grained forwarding control and meter support for implementing rate limits. The reachability of certain resources within the IXP network can be implemented exclusively with certain flow rules installed throughout the network. The destination IP address of the resource can identify flows, which address such a resource (e.g. all packets addressed to

the route server). Such flow rules matching on the traffic towards a certain resource can be assigned with a meter, to implement a specific rate limit. The rates have to be set to a reasonable amount of traffic, to account for events where more traffic to a resource is normal. For example, a higher rate of Border Gateway Protocol (BGP) messages towards the route server happens in case of a router reboot.

In the future, an SDN controller could be an additional resource within the IXP network that is worth securing against attacks on a packet level.

4 IXP Management

In the following section we highlight two SDN use cases for simplifying the IXP management. We believe that SDN developments will bring standardized interfaces that enable configurations of forwarding devices in a centralized manner. Furthermore, we envision SDN to play a key role in future monitoring systems.

4.1 Central Configuration

Current Situation

IXPs attempt to simplify the process of connecting new members to their switching fabric. Thus, the network should allow to manage the addition of new members via a centrally configurable management system. Since large IXPs moved towards a distributed network infrastructure, multiple switches have to be configured at once. While most remote configuration approaches are vendor specific, not all networks are built with hardware from the same vendor. Thus, remote configuration becomes increasingly important even in networks built from different vendor's hardware.

While, OpenFlow is a predominant SDN based protocol to configure the forwarding behavior of a network, it is often confused with a management protocol. Innovative developments in the domain of virtual software switches have brought useful concepts for configuring hardware switches remotely [19]. Thus, we envision SDN as the future unified configuration interface for switches across different vendors.

IXP networks have grown from a single switch to multiple switches distributed over multiple data centers. Since the member of IXPs are usually present in different data centers across a certain area, the IXP tries to expand to those data centers. This eases the effort for a member to connect its infrastructure with the IXP network.

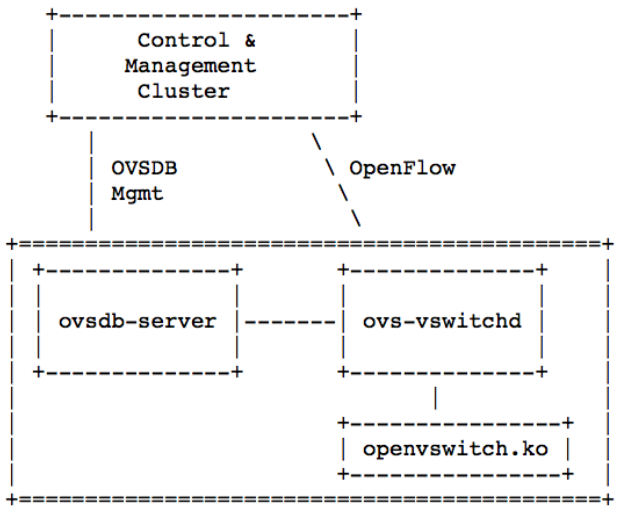

Figure 2: Open vSwitch Interfaces [22].

Among the different tasks, managing an IXP network includes the configuration of individual switches. With the expansion to a distributed infrastructure, IXPs have faced the challenge of how to efficiently manage and configure these switches. A major difficulty came from the fact that there was no unified configuration interface available across different switch vendors. Even today, configuring switches or networking hardware in general still relies on command line interfaces, which are usually not designed to be accessed remotely.

Available Solutions

The challenge of central configuration of a set of distributed networking devices is not limited to IXP networks. Internet Service Providers (ISPs) and enterprises also operate large scale networks, including a huge number of distributed networking devices. Therefore, approaches have been developed to remotely and centrally configure networking devices. YANG [2] is a modeling language used to model configuration and state data for a networking device. The modeled state can be transferred to each device using the Network Configuration (NETCONF) protocol [10]. RESTCONF [1] is another approach to simplify the remote management of network devices, which makes use of both YANG and NETCONF.

Even though these approaches are under development since years, they

still lack sufficient support of hardware vendors. First, not every hardware vendor has yet adopted NETCONF as a unified way to configure their hardware. Second, even if a vendor offers support for NETCONF, he can still implement proprietary interfaces within the NETCONF markup scheme. Therefore, this prevents from achieving a unified configuration interface. The configuration settings offered through NETCONF still differ between the vendor specific implementations.

Vendors such as Alcatel-Lucent have developed their own proprietary systems [2] to centrally manage configuration of multiple devices. Those systems are usually closed source and vendor specific. Therefore, they do not allow managing configuration state between hardware of different vendors. Still, IXPs rely on those systems to develop their own management systems on top. DE-CIX uses a in-house developed system to centrally manage the configuration of each individual switch. For smaller IXPs, which do not have the resource to develop their own systems, IXP-Manager [3] is a common platform to ease the management of IXPs.

Technical Description

Decoupling the control and data plane is one of the key benefits that SDN promises to deliver to the networking community. While the protocols that operate on the control plane (e.g., BGP and OSPF) are well-known and understood, protocols operate on a management plane lacks an extensive study of their properties. Section 4.1 already discusses NETCONF and RESTCONF, which operate on a management plane level. Additionally, the success and adoption of Open vSwitch (OVS)[4] in virtualized server environments brought Open vSwitch Database Management Protocol (OVSDB) [22] to a wider audience. OVS is a virtual switch which purely software based. It was initially developed for virtualized server environments, where it interconnects mutliple Virtual Machines (VMs) on the same host server. OVS consists of two integral parts, a database server holding the configuration state of the virtual switch and a switch daemon which implements the forwarding logic.

For a programmatic access to the OVS database server, OVSDB [22] has emerged as a standardized access protocol. While OVS was designed as a software switch, parts of OVS are reused within hardware switches nowadays. They provide the same interface such as an OVS running on

[2] `https://www.alcatel-lucent.com/products/5620-service-aware-manager`
[3] `https://github.com/inex/IXP-Manager`
[4] `http://openvswitch.org`

commodity hardware. Thus, they allow to access internal configuration state via OVSDB.

Beside decoupling the control plane and outsourcing it to a central controller as envisioned by OpenFlow, SDN also pushes the development of an unified vendor-neutral configuration protocol, such as OVSDB. Note, that OpenFlow is not a management protocol, but rather defines an interface towards the data plane. It allows control plane protocols running on a central controller to instruct a switch how to forward a certain packet, but not to shutdown a certain port, which is part of the management plane. Therefore, protocols such as OVSDB and OpenFlow can be used in conjunction or individually. Figure 2 depicts an overview of the interfaces of OVSDB and their interdependencies.

4.2 Adaptive Monitoring

Current Situation

Continuous operation and early failure detection requires a holistic and flexible monitoring of the entire IXP infrastructure. The traffic rates are monitored per device and interface. However, the current state-of-the-art monitoring systems usually lack the ability to monitor a certain end-to-end path through the IXP distributed switching fabric.This is the case since flow based monitoring are usually only deployed on edge switches. Thus, we expect a rapid development of novel monitoring tools based on the granularity and flexibility SDN can provide. Extended monitoring capabilities will be an integral building block to enable a variety of other use cases discussed in Deliverable 4.3 [8].

Monitoring the current state of the overall network is essential for the detection of failures within the network. A fast failure detection is crucial for implementing appropriate countermeasures and recovering from failures as fast as possible.

The network architects frequently require information about how much peering traffic is exchanged between different members. These statistics must be gathered at different devices within the network. However, they must be stored centrally in order to be processed and evaluated.

Currently available monitoring solutions are capable of providing a snapshot of the overall traffic volume within a network and on individual links. In addition, they can monitor individual flows based on their header information. Besides, it is challenging to identify which actual path a certain packet or flow has taken through the switching fabric including multiple hops. Even though flow-based monitoring tools in principal are able to ac-

complish this, they usually sample traffic in order to cope with higher traffic volumes. Furthermore, flow-based statistics are usually exclusively gathered at the edge of a network, to reduce the storage requirements.

Available Solutions

sFlow [23] is a widely deployed tool for monitoring the data plane of a networking device. sFlow implements sampling based monitoring, where one out of N packets is captured at the switch and then send to a central sFlow collector. The collected data is usually limited to the header information of a packet. Thus the payload is not available to the sFlow collector. Additionally, most switches support sFlow counter, where counter such as transmitted bytes and packets are stored per interface. The current link utilization is estimated by polling those counters periodically.

NetFlow [5]/IPFIX [6] have recently emerged as a new widely adopted standard to capture traffic information within a network. Netflow collects individual packets passing through a switch and clusters them into flows, depending on their source and destination IP address, source and destination port number, and IP protocol number. Thus, statistics are collected and aggregated per flow on the switch itself. The switch periodically exports these flow statistics to a remote host, which collect statistics for multiple switches. Netflow supports customizable templates for its statistics, which makes it far more flexible than sFlow.

Port mirroring is another available solution in which a switch can be configured to duplicate each packet on a certain port and send the duplicate to the mirror port. The mirror port is usually connected to a host that captures all the incoming packets from the mirror port for further inspection. While port mirroring in principal allows for a comprehensive view on the packets send through a certain port, it is a solutional that does not scale well in practice. A single mirror port can only mirror ports which in total do not exceed its available bandwidth depending on their current utilization. In IXP scenarios, where 100G ports become increasingly popular, mirroring for multiple 100G ports is extremely challenging due to the sheer amounts of data to process.

Technical Description

The capabilities of SDN-based monitoring are mainly defined by OpenFlow since it is the most widely used implementation of the SDN southbound interface. OpenFlow defines statistic counters per flow rule. Therefore, it allows the controller to install flow rules network wide at any desired granularity. OpenFlow hardware also keeps track of interface counters, similar

to sFlow counters. By combining interface counters with fine-grained per flow counters, an IXP operator can improve its view of the network state while relying on the interface counter for his bird's eye view on the traffic volume. The ability to monitor certain traffic with a higher granularity is crucial for implementing innovative use case for IXP members, as described in Deliverable 4.3 [8]. Especially, DoS attack detection can benefit from statistics gathered from certain fine-grained flows within the IXP network.

Currently, an OpenFlow controller polls each switch for the per flow counters and interface counters, depending on the polling interval and the number of counters, these operations can easily overwhelm a switch's management CPU [7]. However, recent additions to the OpenFlow standard [20] include push-based statistics. With this concept a switch can automatically send certain counter statistics to the controller upon exceeding a predefined threshold. Push-based statistics requires the controller to carefully assign thresholds to certain flow rules, in order to receive an update on those counters when necessary. This concept can fundamentally change the way IXP networks are monitored today. Instead of frequent polling of all available data and costly processing afterwards, the monitoring task is distributed over the network to each individual networking device. By carefully defining the thresholds, an IXP operator can receive statistic updates on demand.

5 IXP Infrastructure

In this section we introduce two SDN use cases that impact the IXP infrastructure. We exploit the programmability of SDN to enhance load balancing for IXP networks. Furthermore, we describe a simplified label switching concept inspired by the fundamentals of Multiprotocol Label Switching (MPLS).

5.1 Load Balancing

Current Situation

IXP networks are currently facing new challenges driven by the increasing peak traffic values up to five Tbps. The network design has to reserve enough capacity among the members in order to steer this enormous amount of exchanged traffic. Additionally, the network has to cope with a growing number of available ports at its edge switches for connecting member with port speeds of up to 100G. This lead to IXP networks providing up to

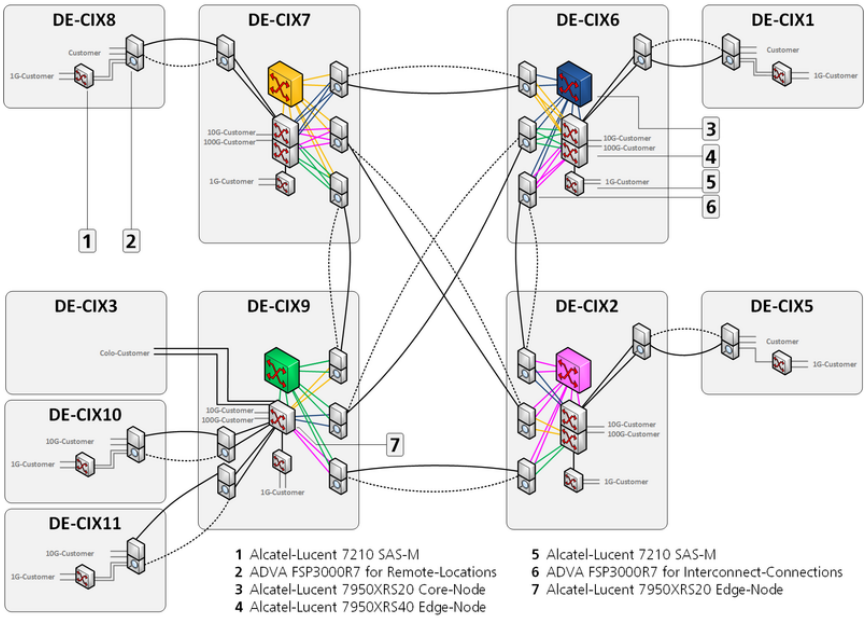

Figure 3: Network topology of DE-CIX Frankfurt.

about 18 Tbps of connected bandwidth, e.g., DE-CIX Frankfurt[5]. Building a resilient network capable of handling these amounts requires a sophisticated design that leverage load balancing mechanisms. While load balancing schemes are already widely deployed in today's IXP networks, we explore in this use case the opportunities of leveraging these schemes with the deployment of SDN-enabled hardware within the IXP.

A number of IXPs (e.g., DE-CIX, AMS-IX) have grown from a single switch infrastructure to large infrastructures with distributed switches in different data centers. One of the main requirements for scaling an IXP infrastructure is a high port, to connect as many member router as possible at the same switch. Beside this scaling challenge at the edge of the IXP network, this enormous traffic growth imposes challenges also within the interconnection network between these edge switches. Larger IXPs such as DE-CIX and AMS-IX have established an additional core layer inside their IXP network that interconnects their edge switches.

The core layer requires a careful bandwidth planning, in order to provide sufficient forwarding capacity to interconnect the edge switches. DE-CIX

[5] https://www.de-cix.net/news-events/latest-news/news/article/peak-data-traffic-at-de-cix-breaks-5-terabit-per-second-record/

simplified this bandwidth planning by operating four equal core switches. The core layer design is depicted in Figure 3. Each of the four high capacity edge switches (one located at DE-CIX 2,6,7, and 9) is interconnected with the same bandwidth to each of the four core switches, located at DE-CIX 2,6,7, and 9 respectively. These links can reach a bandwidth of up to 2.4 Tbps and therefore consist of multiple individual links bundled together as a Link Aggregation Group (LAG).

Even though a portion of the overall traffic remains local at a certain edge switch, a large fraction of the overall traffic traverses the core layer. This large fraction forces network operators to carefully configure their load balancing scheme in order to optimize the load per link within the IXP network. To cope with this large fraction of traffic, DE-CIX uses Equal-Cost Multi-Path Routing (ECMP) [14] to equally spread the load among the four core links. Even though ECMP manages to keep the load for each link in balance, it requires each link to be equal in bandwidth. Therefore, each bandwidth upgrade of a core link requires all three other core links to be upgraded equally. Indeed, this results in a large over provisioning of the whole IXP network in terms of available link bandwidth, which is expensive (mainly CAPEX but also OPEX).

However, in case of a LAG member failure as a subset of a LAG, interconnecting an edge switch with a core switch, the available bandwidth of this particular LAG decreases. ECMP does take such a bandwidth decrease of individual LAGs into account and keeps balancing the traffic equally across all available LAGs. As long as the remaining bandwidth is sufficient for the current traffic volume traversing this LAG, the remaining LAG members can still be used as forwarding links. However, if the traffic volume exceeds the available LAG bandwidth, two possible measures can be taken: (i) the affected LAG can be shut down to avoid any further traffic to traverse it and potentially overload it or (ii) the fraction of balanced traffic that is pushed over this LAG could be reduced. Since the latter is not supported by ECMP, the first measure is applied at DE-CIX, leading to a waste of precious bandwidth.

In addition, ECMP spreads traffic in a static way (i.e., using an hashing algorithm), without the ability to obtain any feedback on how much bandwidth of a certain LAG is used. This information would be valuable for further tweaking the load sharing mechanism, leading to a dynamic load-balancing mechanism. This is especially important, if load balancing the traffic to a certain core switch should also take the available bandwidth of this core switch to the final edge switch into account. Currently, ECMP does not consider the bandwidth of this second hop.

Available Solutions

Spreading traffic across links with different bandwidths is a common challenge for multi-path networks, such as ISP networks. Therefore, current approaches aim to extend ECMP to support weighted-load-balancing [30]. In these approaches ECMP can be configured to spread traffic non-equally among a number of links, e.g., to allocate more traffic to a link with a higher capacity. Nevertheless, these approaches are not yet widely deployed. The hardware installed at DE-CIX Frankfurt also lacks support for weighted ECMP especially for LAGs consisting of multiple 100G links. If the actual implementation of ECMP is restricted to equal load balancing, shutting down a complete LAG after the failure of a certain number of LAG members is the only practical available solution. This further emphasizes the need for over provisioning of LAGs, in order to keep them operational even during a failure of individual LAG members.

Technical Description

A more sophisticated load balancing approach should aim for two goals. First, it should allow IXP operators to spread traffic non-equally among certain links, while taking the available bandwidth of each individual link into account. Second, it should provide extended visibility to cover a complete path through an IXP network. This yields visibility of all link loads on an end-to-end path.

SDN offers two potential benefits for implementing such a sophisticated load balancing approach. The flow-based programmability of the forwarding plane allows novel load balancing schemes [31]. While still relying on hash algorithms for distributing traffic across different output ports, introducing multiple flow rules to balance traffic among different output ports enables more flexibility and control. It is worth noting that distributing traffic among different output ports using OpenFlow requires the group type select specified in the OpenFlow standard [20]. This feature is optional and therefore not necessarily supported by every OpenFlow-enabled hardware switch.

Additionally, SDN features a central controller, which has a global view of the network topology. Therefore, it becomes much easier to gather information such as link utilizations on one end of the network. It allows network operators to use this information to control the forwarding behavior at another part of the network. In the current topology adopted by DE-CIX, we could use a controller to collect link utilization leveraging flow or interface counter of all edge to core switch links. When spreading traf-

fic across the four available core switches, the utilization of the core to final edge switch links can be taken into account. This reduces the need for heavy over-provisioning bandwidth within the IXP network.

Operating links at utilization close to 90% and above requires flexibility within the network to react on traffic patterns by changing the forwarding behavior of certain flows if needed. Jain et al. [16], have accomplished this flexibility with their SDN Wide Area Network (WAN) deployment with a central SDN controller. Therefore, we believe that an SDN deployment within an IXP network enables more control and flexibility in terms of load balancing the amounts of traffic exchanged over those networks today.

Furthermore, the SDN's fine-grained forwarding scheme allows for any network topology without being forced to build highly symmetric topologies because ECMP requires so.

5.2 Layer 2 Label Switching

Current Situation

Larger IXPs moved towards a layer 3 based infrastructure, e.g., MPLS, emulating a layer 2 service. This shift was required in order to efficiently leverage multiple paths inside their infrastructure to accomplish both scalability issues and increase resilience. While MPLS requires a underlying layer 3 network, it comes at the cost of increased complexity for design and operation of an IXP network. For this use case we investigate the opportunities of SDN to develop a simplified layer 2 label switching concept, which can reduce the protocols employed in today's IXP environments. Such a simplified concept not only promises a larger IXP with existing experiences in operating an MPLS network, but also enables smaller IXPs to benefit from a greater resilience and simplified operation.

IXP networks greatly vary in size, with smaller IXPs deploying only a single switch and larger IXP networks built on top of multiple distributed switches. In any case, all of them offer a layer 2 transport service to their members. The larger IXP networks, such as the one deployed at DE-CIX, are designed focusing on resilience and scalability in both number of available member ports and backbone bandwidth capacity. These properties are hard to achieve with a pure layer 2 network design. In particular, resiliency is a challenge for growing IXP networks exchanging multiple Tbps. Since layer 2 switching lacks support for an efficient multi-path forwarding, building a resilient network infrastructure usually requires hot-standby components. Hot-standby components only become active in a failure scenario. Therefore their switching capacity can not be used for normal operation. In order to

build and operate an IXP network at reasonable CAPEX, traffic should be distributed among all available switches, including the standby components.

Layer 2 switching is not sufficient for implementing the desired multi-path forwarding that distributes traffic load among multiple switches. It is worth noting that layer 2 networks are inherently limited to single path forwarding by the Spanning Tree Protocol (STP). However, recent standards such as IEEE 802.1aq [12] and TRILL [21] are emerging to replace STP and support multi-path within a layer 2 infrastructure. Both standards have issues when it comes to interoperability between the implementations of different vendors. Additionally, since both are relatively new concepts in comparison to MPLS, they lack sufficient experience and know-how both from vendors and the networking community.

Available Solutions

In order to implement an infrastructure with the characteristics described above, larger IXPs have moved from previously pure layer 2 network infrastructures to a more advanced and flexible Virtual Private LAN Services (VPLS) [17]/MPLS [27] based infrastructure. MPLS is used to implement both resilience and scalability. The core layer design depicted in Figure 3 exploits the ability to load balance traffic across all available core switches based on MPLS label switching and ECMP. VPLS operates on top of this layer 3 network, in order to emulate a layer 2 network behavior.

For a more flexible forwarding of traffic across the IXP infrastructure, MPLS and VPLS lead to an increasing complexity for the network operator as well as for the networking devices. While the use of VPLS is transparent to IXP members, it requires each device within the IXP network to support both VPLS and MPLS. The challenges of operating an MPLS based network are partly because of the enormous feature set of the MPLS protocol. While a large number of these features are beneficial for ISP networks, IXP networks only require a subset. Thus, IXP networks can be implemented with a lighter version of label switching, without the overhead added because of the need of layer 3 routing, e.g., MPLS label distribution.

Technical Description

The goal of this use case is to implement a simplified label switching concept without the complexity of MPLS. It should require less protocol and management overhead within the IXP network. Label switching concepts have already been implemented using OpenFlow [15, 25]. They both facilitate the central OpenFlow controller to maintain the label and path information. Similar to these concepts, we can implement a label switching

concept based on the match and action structure introduced by OpenFlow.

At the edge switches of an IXP network, the ingress traffic is matched with the installed flow rules. These flow rules can be crafted by the controller to match the traffic at a certain granularity (e.g., per member). Each of these flow rules will push a certain label to the packet by either pushing an MPLS label by itself, or by encoding label information into a different header field (e.g., destination MAC). Since the MPLS push and pop operations defined in OpenFlow are not widely supported by the available hardware, we would privilege a label encoding within a different header field, e.g., the destination MAC. Push and pop operations would be then implemented within the ENDEAVOUR SDN architecture, which also needs to ensure to rewrite all header fields at the egress port.

The flow rules installed in intermediate switches between two edge switches only implement matching on the predefined labels. Since the flow rules are centrally installed by a controller, the labels can be globally unique per path. Unlike MPLS, which has to maintain local labels per switch.

Packets arriving at the egress switch of a path are again matched with their specific assigned label, which is removed or rewritten by the OpenFlow action. Thus, the label switching process within the IXP network is transparent to a member.

Implementing a label switching concept with OpenFlow can also solve some inherent limitations of current OpenFlow hardware. Rapidly installing and modifying flow rules within a OpenFlow hardware switch is costly. Therefore, the hardware poses limitations in terms of the number of flow rules it can modify and install. Given the static nature of labelled paths within an IXP network, the flow rule matching on a certain label in intermediate switches is rather static. The complexity of modifying the forwarding behavior of packets along different paths (e.g., for load balancing purposes) remains at the edge switches. A single flow rule modification at such an edge switch is sufficient to change the forwarding behavior of all matching packets along a end-to-end path. This is especially important in case of infrastructure failures, since it allows for fast rerouting of packets along alternative paths.

If the OpenFlow actions for pushing and popping MPLS labels are supported by the concrete hardware, an IXP only requires OpenFlow hardware to be deployed at as edge switches while relying on non-OpenFlow hardware for its core layer. The core layer can simply forward packets based on the MPLS labels inserted by the edge switches.

6 Summary

In this document we collected seven use cases, which show the potential of SDN in simplifying, securing and enhancing operations at an IXP. It clearly shows that SDN does not only bring benefits for novel member features, but also provides advantages to the operator's businesses.

We classified these seven SDN use cases in three main categories: Safety and Security, IXP Management and IXP Infrastructure.

We presented SDN solutions for providing IXP operators capabilities with secure solutions for their IXP network based on access control mechanisms.

Furthermore, we identified broadcast packet handling as an imminent scalability burden for member routers at the scale of the large layer 2 networks that IXPs operate today. We described multiple concepts that all can mitigate the burden of handling large numbers of broadcast packets from the routers.

Additionally, we foresee SDN based solutions for securing network resources within an IXP network (e.g. route servers). These resources are mission critical and therefore require appropriate security measures at the network layer.

In order to provide extended programmability of networking devices, SDN advocates for standardized interfaces to access these devices remotely via software. Thus, we see SDN as an ideal movement towards vendor independent interfaces, which eases the central configuration and programming of distributed networking devices.

Likewise, we see potential in the extended programmability for implementing novel load balancing extensions that specifically address the needs of large IXP network operators. To implement such load balancing extensions, we exploit the fine-grained monitoring capabilities of SDN. Those capabilities are also beneficial for a more flexible and fine-grained monitoring of the IXP infrastructure.

Based on the wide deployment of MPLS at large IXP networks, we envision SDN to enable a simplified version of MPLS, while retaining most of its benefits (e.g. multi-path routing). Our concept foresees a simplified label switching concept for layer 2 networks.

7 Outlook

ENDEAVOUR supports innovation and development at IXPs and therefore at the core of the Internet. Introducing SDN will allow IXPs to innovate at a higher frequency than today. One critical advantage is the increased control over the software stack of their networks. While this innovation will enable IXPs to develop innovative and novel features for their members, it will also lead to a simplified overall IXP operation.

With insights into the operation of a large IXP such as DE-CIX, ENDEAVOUR will further work on fostering incentives for IXP operators to deploy SDN. We will work on implementing the most appealing use cases as a prototype in order to show their potential for the IXP community in practice.

8 Acronyms

SDN Software Defined Networking

BGP Border Gateway Protocol

ISP Internet Service Provider

IXP Internet eXchange Point

AS Autonomous System

IP Internet Protocol

IPv4 Internet Protocol version 4

OSPF Open Shortest Path First

STP Spanning Tree Protocol

DoS Denial of Service

VPLS Virtual Private LAN Services

VM Virtual Machine

EVPN Ethernet Virtual Private Network

WAN Wide Area Network

ARP Address Resolution Protocol

ND Neighbor Discovery

ACL Access Control List

ECMP Equal-Cost Multi-Path Routing

LAG Link Aggregation Group

OVSDB Open vSwitch Database Management Protocol

OVS Open vSwitch

MPLS Multiprotocol Label Switching

NETCONF Network Configuration

MAC Message Authentication Code

STP Spanning Tree Protocol

ICMPv6 Internet Control Message Protocol Version 6

References

[1] A. Bierman, M. Bjorklund, and K. Watsen. Internet-Draft: RESTCONF Protocol, 2015.

[2] M. Bjorklund. RFC 6020: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), 2010.

[3] V. Boteanu. Minimizing ARP traffic in the AMS-IX switching platform using OpenFlow, 2013.

[4] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *ACM IMC*, pages 242–253. ACM, 2009.

[5] B. Claise. Internet-Draft: Cisco systems NetFlow services export version 9. 2004.

[6] B. Claise, B. Trammell, and P. Aitken. RFC 7011: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, 2013.

[7] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee. DevoFlow: Scaling Flow Management for High-performance Networks. *SIGCOMM Comput. Commun. Rev.*, 41(4):254–265, 2011.

[8] C. Dietzel, S. Bleidner, G. Kathareios, P. Owezarski, S. Abdellatif, M. Chiesa, M. Canini, and Antichi. Design of Use Cases for Members of IXPs, 2016.

[9] M. Dittmar. ARP/ND handling with VPLS, 2013.

[10] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman. RFC 6241: Network Configuration Protocol (NETCONF), 2011.

[11] EURO-IX. European Internet Exchange Association. `https://www.euro-ix.net/`.

[12] D. Fedyk and M. Seaman. 802.1aq - Shortest Path Bridging, 2012. `http://www.ieee802.org/1/pages/802.1aq.html`.

[13] G. Hankins. Peering Observations 2007 vs. 2015, 2015. `https://www.peering-forum.eu/system/documents/55/original/20150921_0900_greg_hankins_epf-10-peering-observations.pdf`.

[14] C. Hopps. RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm, 2000.

[15] A. Iyer, V. Mann, and N. Samineni. SwitchReduce: Reducing switch state and controller involvement in OpenFlow networks. In *IFIP Networking Conference, 2013*, pages 1–9, May 2013.

[16] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hlzle, S. Stuart, and A. Vahdat. B4: Experience with a Globally-deployed Software Defined Wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 3–14, New York, NY, USA, 2013. ACM.

[17] M. Lasserre and V. Kompella. RFC 4762: Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling, 2007.

[18] 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.

[19] R. Narisetty, L. Dane, A. Malishevskiy, D. Gurkan, S. Bailey, S. Narayan, and S. Mysore. OpenFlow Configuration Protocol: Implementation for the of Management Plane. In *Research and Educational Experiment Workshop (GREE), 2013 Second GENI*, pages 66–67, Mar. 2013. bibtex: 6601418.

[20] ONF. OpenFlow Switch Specification Version 1.5.1, 2015. `https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf`.

[21] R. Perlam, D. Eastlake, D. Dudd, S. Gai, and A. Ghanwani. RFC 6325: Routing Bridges (RBridges): Base Protocol Specification, 2011.

[22] B. Pfaff and B. Davie. RFC 7047: The Open vSwitch Database Management Protocol, 2013.

[23] P. Phaal, S. Panchen, and N. McKee. RFC 3176: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, 2011.

[24] J. Rabadan, S. Sathappan, K. Nagaraj, W. Henderickx, G. Hankins, T. King, and D. Melzer. Internet-Draft: Operational Aspects of Proxy-ARP/ND in EVPN Networks, 2015.

[25] R. Ramos, M. Martinello, and C. Esteve Rothenberg. SlickFlow: Resilient source routing in Data Center Networks unlocked by OpenFlow. In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pages 606–613, Oct. 2013.

[26] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger. Peering at peerings: On the role of IXP route servers. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 31–44. ACM, 2014.

[27] E. Rosen, A. Viswanathan, and R. Callon. RFC 3031: Multiprotocol Label Switching Architecture, 2001.

[28] A. Sajassi, R. Aggarwal, N. Bitar, A. Isaac, J. Uttaro, J. Drake, and W. Henderickx. RFC 7432: BGP MPLS-Based Ethernet VPN, 2015.

[29] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hlzle, S. Stuart, and A. Vahdat. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. *SIGCOMM Comput. Commun. Rev.*, 45(5):183–197, Aug. 2015.

[30] J. Zhang, K. Xi, L. Zhang, and H. Chao. Optimizing Network Performance Using Weighted Multipath Routing. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–7, July 2012.

[31] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat. WCMP: Weighted Cost Multipathing for Improved Fairness in Data Centers. In *Proceedings of the Ninth European Conference on Computer Systems*, EuroSys '14, pages 5:1–5:14, New York, NY, USA, 2014. ACM.