ENDEAVOUR: Towards a flexible software-defined network ecosystemD4.6: Final Report about Tests

Daniel Kopp, Marco Canini, Marco Chiesa, Christoph Dietzel, Eder Leao Fernandes, Rémy Lapeyrade, Philippe Owezarski, Matthias Wichtlhuber

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	4.6
Document title	Final Report about Tests
Document version	1.0
Editor in Chief	Kopp, DE-CIX
Authors	Kopp, Canini, Chiesa, Dietzel, Fernandes, Lapeyrade, Owezarski, Wichtlhuber
Date	22/12/2017
Reviewer	Gusat, IBM
Date of Review	21/12/2017
Status	*Public*

Revision History

Date	Version	Description	Author
27/11/17	0.1	First structure	Kopp, Dietzel
10/12/17	0.2	Inbound/Outbound TE added	Chiesa
10/12/17	0.3	Access Control added	Chiesa
11/12/17	0.4	Broadcast Prevention added	Fernandes
12/12/17	0.5	Anomaly Detection added	Lapeyrade
15/12/17	0.6	Summary added	Kopp
19/12/17	0.7	Review	Canini
20/12/17	0.8	Typos corrected	Owezarski
20/12/17	0.9	Review	Dietzel
21/12/17	1.0	Final Review	Gusat

Executive Summary

To evaluate the correctness of the ENDEAVOUR platform's implementation we conducted recurring test by using a set of well-defined test cases. In all major stages of the development process, the use cases have been tested for their functional correctness. Extensive testing and structured development within the virtual environment enabled an straight forward transition of the implemented use cases onto the hardware testbed. Moreover, additional test and examinations on a variety of performance and scaling aspects proved the ability of the ENDEAVOUR platform to be deployed in a real wold scenario and provides valuable insights to the research community.

Contents

1 Introduction

This report describes the results and final tests with the implementation of ENDEAVOUR. It is use cases- and feature-centric, but also reports about the general development and test design.

Within all phases of the project, a recurring development cycle was consistently followed. At first, the core functions have been implemented and tested. With the successful completion of basic functionalities, the development of the use cases of the ENDEVOUR platform started. Whenever we found that a component under development required additional adoption of basic functionalities, it was fed back for implementation and testing.

Most use cases passed two milestones, which have been the implementation of all individual features in the virtual testbed and, in a second step, their integration and deployment to the hardware testbed. Besides functional tests, specific examinations of the performance of components and critical features were conducted. Moreover the Software Defined Networking (SDN) hardware switches that are used within the testbed have been tested for their performance and compatibility.

2 ENDEAVOUR Testbeds

This Section briefly describes the characteristics and topologies of the virtual and hardware testbed used for implementation and testing of the ENDEAVOUR platform.

2.1 Virtual Testbed

The virtual testbed comprises a collection of virtual machines (VM) customized to emulate the behaviors of real switches and routers. The topology of the virtual testbed is established with the MiniNet[1] emulation tool, which is installed during the VM provisioning operation. Each router in MiniNet runs the zebra and bgpd daemons, part of the Quagga routing engine.[2]

We use the network topology depicted in Figure 1 to demonstrate the ENDEAVOUR platform on the virtual testbed. Three Internet eXchange Point (IXP) members `A`, `B`, and `C` connect to an ENDEAVOUR IXP fabric, which consists of a Core-Edge topology with 4 core switches and 4 edge switches. Member `A` owns a Border Gateway Protocol (BGP) border

[1] `http://mininet.org/`
[2] `http://www.nongnu.org/quagga/`

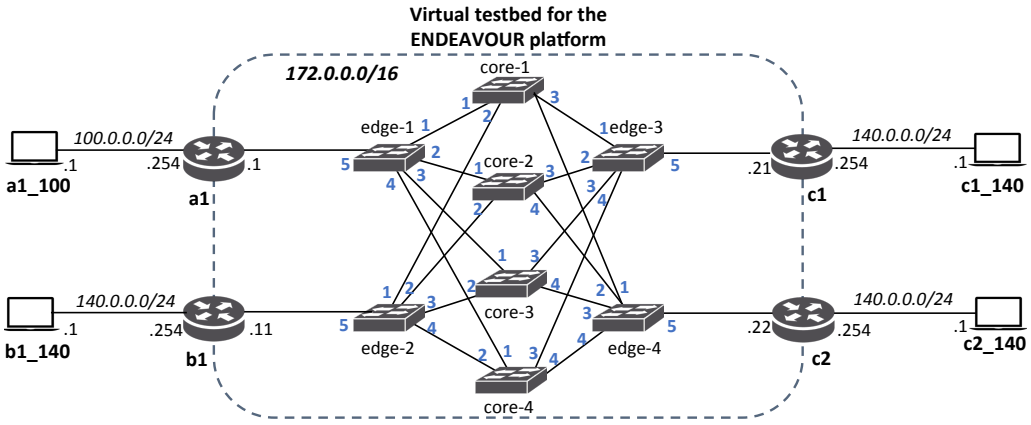


Figure 1: The ENDEAVOUR virtual testbed.

router `a1` connected to `edge1`, member `B` owns a BGP border router `b1` connected to `edge2`, and Member `C` owns two BGP border routers `c1` and `c2` connected to `edge3` and `edge4`, respectively. A host `a1_100` with Internet Protocol (IP) address `100.0.0.1` is connected to member `A`'s router and three hosts `b1_140`, `c1_140`, and `c2_140` with the same IP address `140.0.0.1` are connected to routers `b1`, `c1`, and `c2`, respectively. The IXP IP subnet is `172.0.0.0/16` and the exact address of each member's border router is depicted in the figure close to the member's router. Member `A` announces `100.0.0.0/24` while members `B` and `C` both announce `140.0.0.0/24`. The port numbers used within the IXP fabric are highlighted in blue.

2.2 Hardware Testbed

To test the feasibility and scalability of the ENDEAVOUR architecture in real-world settings and with real switches, we created a hardware testbed, which we deployed at one of the DE-CIX data centers. The testbed consists of three switches and four servers, interconnected as shown in Figure 2.

The 4 servers are `sv-01`, `sv-02`, `sv-03`, and `sv-08`. The participant network devices are executed on `sv-01`, `sv-02`, and `sv-08` through `docker` containers. The ENDEAVOUR controller, the Route Server relay, and the Address Resolution Protocol (ARP) proxy are executed on `sv-03`.

Three hardware switches are used in the testbed: `novi-switch`, `edge-core`, and `centec`. We decided to implement 2 edge switches by partitioning the ports belonging to the `novi-switch` switch because the other switches were

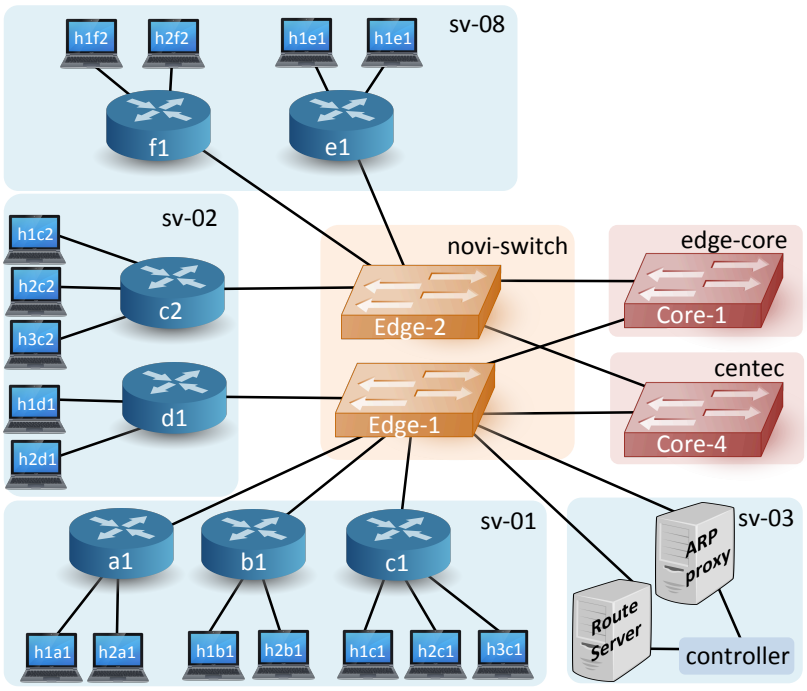


Figure 2: The ENDEAVOUR hardware testbed.

not suited for supporting the edge forwarding functionality (described in Deliverable 2.2). We used the `centec` and `edge-core` switches as the core switches of the IXP fabric. The final testbed interconnection network is shown in Figure 2, which consists of an edge-core IXP topology with two edge switches interconnected to both core switches. We note that the German Commercial Internet Exchange (DE-CIX) topology is also an edge-core topology.

Table 1: Overview of Tests

Use Case	Criteria	Virtual	Hardware
Inbound/Outbound TE	4.1-A	✓	✓
	4.1-B	✓	✓
	4.1-C	✓	✓
	4.1-D	✓	✓
	4.1-E	✓	✓
	4.1-F	✓	✓
Advanced Blackholing	4.2-A	✓	✓
	4.2-B	✓	✓
	4.2-C	✓	✓
	4.2-D	✓	✓
	4.2-E	✓	✓
Traffic Anomaly Detection	4.3-A	✓	-
	4.3-B	-	✓
	4.3-C	-	✓
	4.3-D	-	✓
Broadcast Prevention	5.1-A	✓	✓
	5.1-B	✓	✓
	5.1-C	✓	✓
Access Control	5.2-A	✓	✓
	5.2-B	✓	✓
	5.2-C	✓	✓
	5.2-D	✓	✓
Load Balancing	5.3-A	✓	✓
	5.3-B	✓	✓

3 Test Design

To evaluate the correctness of the ENDEAVOUR platform's implementation of the use cases on the virtual and hardware testbeds, we designed a set of test cases. For each use case, we provide context to understand the benefits and scope of the use cases first. Then, we describe how each test is designed and executed.

To assure that our platform runs as intended, we designed a set of `acceptance criteria` for each use case. This ensure a high level of quality. Each set of acceptance criteria consists of statements, each with a clear pass or fail result, that specifies requirements, and is applicable for each use case.

All results were reproduced at least 3 times.

Finally, we report on the results of all tests for each use case. In cases where not all defined acceptance criteria could be met, we discuss the reasons and sketch possible solutions.

For the implemented member use cases (demonstrated in Deliverable 4.7), namely Inbound/Outbound TE, Advanced Blackholing, and Traffic Anomaly Detection, the corresponding tests can be found in Section 4. The operator use cases (demonstrated in Deliverable 4.8), i.e., Broadcast Prevention, Access Control, Load Balancing, are summarized in Section 5. Table 1 summarizes the results of all use cases.

4 Implemented Member Use Cases

This section describes the tests including the results for the implemented member use cases [6].

4.1 Inbound/Outbound TE

TE, i.e., the task of tuning routing protocol parameters so as to optimize traffic flows, is a fundamental and crucial operation in today's network. Given the rich and flourishing connectivity ecosystem at IXPs, operators wish to carefully control how traffic enters/leaves their networks with the ultimate goal of enhancing network performance. To this end, the ENDEAVOUR platform is designed to support advanced fine-grained Inbound/Outbound TE.

The implementation of fine-grained routing capabilities is the main part of the industrial Software-Defined-Exchange (iSDX) component. We refer the reader to Deliverable 2.2 for a detailed description of the ENDEAVOUR architecture and, in particular, the iSDX component. In addition, we refer the reader to Deliverable 2.3 to complement the architecture description with a detailed low-level explanation of the iSDX encoding mechanism at the forwarding-plane level. The description of the content of the forwarding tables has been thoroughly described in Deliverable 2.3 with the exception that, for the hardware testbed, all tables have been shifted by one as table 0 of the NoviFlow switch does not allow to match on Layer 4 fields.

Test Description The Inbound/Outbound use case demonstrator is built upon the hardware testbed at DE-CIX using simplified IXP scenario depicted in Figure 3. Member `A` wishes to send HTTP traffic towards member

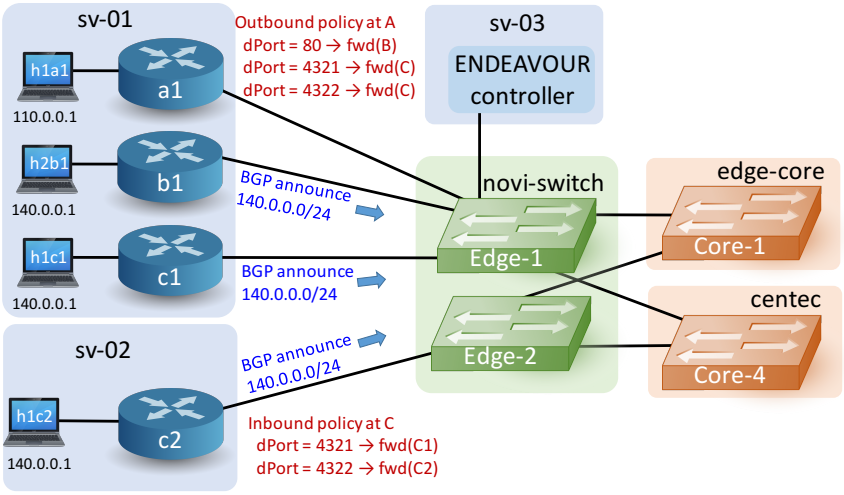


Figure 3: ENDEAVOUR hardware testbed topology for the Inbound/ Outbound TE use case.

`B` and traffic destined to ports `4321` and `4322` towards member `C`, who, in turn, aims at steering this incoming traffic towards ports `C1` and `C2`, respectively.

The demonstration evolves as a sequence of 6 phases:

1. The ENDEAVOUR platform is started. The outbound and inbound policies are installed into the NoviFlow edge switch. The Route Server component of the ENDEAVOUR platform receives the BGP announcements and creates the necessary Virtual IP next-hops and Virtual Media Access Control (MAC) addresses that are sent to member `A` with a gratuitous ARP reply.

2. Host `110.0.0.1` generates a flow of HTTP traffic towards `140.0.0.1`.

3. Member `B` withdraws its BGP announcement for the IP subnet `140.0.0.0/24`. At the same time, member `A` generates another 60sec. HTTP traffic flow towards `140.0.0.1`.

4. Member `B` re-announces a BGP announcement for the IP subnet `140.0.0.0/24`. At the same time, Host `100.0.0.1` generates another one-minute HTTP traffic flow towards `140.0.0.1`.

5. Host `100.0.0.1` generates a 60sec. traffic flow towards `140.0.0.1` destined to port `4321`.

6. Host `100.0.0.1` generates a one-minute traffic flow towards `140.0.0.1` destined to port `4322`.

Acceptance Criteria The test scenario that was used at all stages of the development consisted of the following individual criteria and test details:

4.1-A The ENDEAVOUR platform installs the forwarding state into the Outbound and Inbound forwarding tables that reflects the Outbound and Inbound policies of members `A` and `B`, respectively.

4.1-B The first flow of traffic generated by member `A` destined to port `80` is correctly received by member `B`.

4.1-C The second flow of traffic generated by member `A` destined to port `80` is correctly received by member `C` as member `B` has withdrawn its BGP announcement for `140.0.0.0/24`.

4.1-D The third flow of traffic generated by member `A` destined to port `80` is correctly received by member `B` as member `B` re-announces a route towards `140.0.0.0/24`.

4.1-E The flow of traffic generated by member `A` destined to port `4321` is correctly received by member `C` through port `C1`.

4.1-F The flow of traffic generated by member `A` destined to port `4322` is correctly received by member `C` through port `C2`.

Results from Virtual Testbed Results from the virtual testbed have been extensively described in Deliverable 4.5. All the acceptance criteria were satisfied.

Results from Hardware Testbed All the acceptance criteria were satisfied.* The forwarding state has been installed exactly as described in Deliverable 2.3 (more details in the demonstrator video provided below). We monitored traffic using the Monitoring table of the ENDEAVOUR platform. The 5 phases related to the last 5 acceptance criteria are depicted in Figure 4a, where we use different colored lines to draw the different type of traffic entering the IXP network from member `A`. Figure 4b shows how traffic is being received by members `B` and `C`. We can observe that HTTP traffic

(a) Traffic originated from `A1`.

(b) Traffic received at `B`, `C1`, and `C2`.

Figure 4: Demonstration of the Inbound/Outbound TE use case.

is correctly being received by member `B` whenever this one announces an IP prefix towards `140.0.0.0/24` via BGP (i.e., phases 1 and 3). The same HTTP traffic is re-routed through member `C` when member `B` withdraws its BGP announcement for `140.0.0.0/24` (i.e., phase 2). Finally, during the last two phases, we can observe that traffic destined to ports `4321` and `4322` is correctly being transferred via member `C`'s border routers `C1` and `C2`, respectively. A video demonstrating this of this use case in the hardware testbed can be found in Deliverable 4.7.

4.2 Advanced Blackholing

While the relevance of the Internet grew steadily, it also attracts increasingly more cyber-attackers and bad actors. As one example, Distributed Denial of Service (DDoS) attacks are designed to exhaust resources of the targets, e.g., bandwidth or computing power, and disrupt a service on the Internet. Consequently, the number of counter-measures and mitigation service providers is flourishing. With respect to mitigation techniques for the Internet backbone, blackholing is a common mechanism among IXPs and Internet Service Providers (ISPs) [10]. It allows Autonomous Systems (ASes) to discard unwanted traffic at inter-domain boarders before it reaches the destination. In the event of a DDoS attacks, an IXP member can block specific traffic flows sent to his IPs or IP prefixes [7]. Blackholing is considered as a last resort to relieve infrastructure under stress and rather sacrifice reachability for a subset of the Internet to maintain the ongoing service of the more important ones.

For commodity blackholing, which is state-of-the-art, only filtering rules for specific destination IPs combined with the source AS can be defined. This

method provides for limited options in how to define rules, and offers only coarse-grained filtering. Especially, given the nature of DDoS attacks that often use or target specific applications ports, a more detailed set of blackholing rules would provide for a more effective usage of blackholing [5]. Moreover, commodity blackholing builds on BGP for signaling of rules, which limits its use to the more technically versed IXP members. In contrast to commodity blackholing, the ENDEAVOUR platform provides advanced blackholing capabilities, while still remaining backward compatible. The advanced blackholing features of the ENDEAVOUR platform consists of an API and a fine-grained set of rules to define traffic flows to be discarded. By providing more fine-grained rules ENDEAVOUR extends the basic features of commodity blackholing with the most important Internet protocol attributes, e.g., src/dst port, src/dst MAC address, and src/dst IP address. The Application Programming Interface (API) allows IXP members to define, activate, disable, and monitor blackholing rules. This increases the usability of blackholing and provides a well-known interface. Furthermore, it opens the possibility for an IXP to implement a web portal or enable IXP members to easily automate and integrate blackholing in their already existing DDoS solutions.

Test Description During the development of the ENDEAVOUR platform all functional features of the advanced blackholing use case have been tested multiple times. Tests have been conducted once in the virtual testbed at an earlier stage and finally on the hardware testbed at the end of the deployment process. In both environments the implemented features of the use case have been tested successfully.

To ensure the correct function of all features of the advanced blackholing use case, a simplified scenario of an DDoS attack is used. All tests took place in an exemplary IXP setup with several connected member networks and hosts. For the virtual testbed the network was established as depicted in Figure 1 and for the hardware testbed the architecture is described in Figure 2. A number of network flows are started across the different members, to mimic a realistic operation of an IXP. Among them there are specific flows that represent a DDoS attack and target one IXP member. Several rules are installed with the advanced blackholing feature to discard these specific traffic flows. The monitoring feature of ENDEAVOUR is used to observe effectiveness and correct behavior. To verify the observation of the monitoring, measurement software like tcpdump at the DDoS targets can be used. While installing blackholing rules, we observe that all other flows re-

mained unaffected. Finally the blackholing rules are deleted. This completes the test scenario and ensures that all features of the advanced blackholing are functional and effective.

Acceptance Criteria The test scenario that was used at all stages of the development consisted of the following individual criteria and test details:

4.2-A A backholing rule, which specifies source and destination IP addresses, has to be installed within testbed. This rules has to terminate a former running traffic flow. The correct function is verified by the ENDEAVOUR monitoring and on the destination host.

4.2-B A blackholing rule, which makes use of the protocol type, has to be installed and monitored for correct function. The correct function is verified by the ENDEAVOUR monitoring and by recording the traffic flows on the destination host.

4.2-C Blackholing rules with specific protocol ports have to be installed and monitored for correct execution. This can also be verified by the ENDEAVOUR monitoring and on the destination host.

4.2-D All installed blackholing rules have to be deleted. The success of this test is verified by observation of the destination hosts and at the ENDEAVOUR monitoring.

4.2-E While specific blackholing rules are installed, all other traffic have to continue unaffected. The correct behavior is verified by the ENDEAVOUR monitoring and on the destination host.

Results from Virtual Testbed The initial development of the software for the advanced blackholing use case have been tested in the virtual testbed. Data traffic generation for the test scenario was done by using a test framework named *torch*. With the initial development phase, numerous feedback loops between this use case and fundamental components of ENDEAVOUR had been passed. Every iteration was tested until all acceptance criteria where meet and all features have been implemented. A demonstration of the blackholing use case in the virtual testbed can be found in Deliverable 4.4.

Results from Hardware Testbed Within this stage of the development process, all ENDEAVOUR components from the virtual environment have been deployed to the hardware testbed. The correct functioning of the advanced blackholing use case was verified by using the test scenario. Traffic flow generation within the hardware testbed was be done with *iperf* instead of *torch*. All criteria without 4.2 where running without adjustments. To enable to enable 4.2 and allow for specific match on User Datagram Protocol (UDP) and Transport Control Protocol (TCP) ports, the forwarding needed to be reconfigured due to the difference in the hardware switches. Finally all criteria could be fulfilled and with this all features of the advanced blackholing use case where supported by the hardware testbed. This marked the end of the development of this use case. For further information about the test description and use case demonstration, the reader is refered to Section 5. A demonstration of the blackholing use case presented in the hardware testbed can be found in Deliverable 4.8.

4.3 Traffic Anomaly Detection

IXPs networks forward traffic from various member of different sizes and origins. Anomalies (attacks included) can be produced by one of such members, and be forwarded by the IXP to another member. Considering the position of the IXP in this case, collaboration between members and IXP operator can occur in order to inspect traffic anomalies. Depending on the results of the anomaly detection software, the member can take immediate actions for its received traffic. One of the actions can be applying traffic blocking techniques, as described in subsection 4.2.

However, the deployment of such solution needs to be compatible with the current trend in IXPs on traffic per member. This solution is intended to be offered as an individual service per member.

Test Description The following tests have been achieved in two steps. First, the use case has been tested in order to showcase a Proof of Work (PoW) of the entire solution. Then, we performed several tests to indicate if the entire solution is suitable for an IXP member traffic rate. This implies the evaluation of our solution on two traffic metrics: packet rate and flow rate.

To simplify the test procedure, we will be focusing on generating realistic traffic from one member. The traffic is mirrored to the OSNT card, and then processed by an anomaly detection software (ORUNADA [8]). Based on

the anomaly reports delivered by this software, we compare against the list of attacks injected into the traffic member.

Acceptance Criteria The following acceptance criteria ware created to validate on a first step the viability of the solution in a virtual environment, then reassuring the main concerns of scalability on the hardware deployment.

4.3-A The anomaly detection software is able to detect the set of attacks on an offline traffic capture file, after the OSNT card processing step.

4.3-B The full hardware implementation is able to detect the set of anomalies inserted in the member traffic network.

4.3-C There are no significant performance issues when the member's traffic is increasing on the packet per second rate.

4.3-D There are no significant performance issues when the member's traffic is increasing on the flow rate.

Results from Virtual Testbed The first criteria was validated in a virtual testbed. However, this testbed could not handle the member traffic requirements. The hardware processing capabilities of the OSNT is required. There is no emulation platform for handling the same functionalities as the card - i.e., precise timestamps and packet features extraction. Therefore, on the scaling criterias and hardware implementation, we could not validate them in a virtual testbed. A description of this use case and a video demonstrating it can be found in Deliverable 4.5.

Results from Hardware Testbed The first test didn't fulfill the acceptance criteria due to hidden bottlenecks on both the implementation of OSNT and ORUNADA. After increasing the performance of both components, the acceptance criteria were validated in our hardware testbed. The final results of the set of last 3 tests can be found in the D3.4 [9]. A video demonstrating the final implementation of this use case can be found in Deliverable 4.8.

5 Implemented Operator Use Cases

This section describes the tests including the results for the implemented operator use cases [3].

5.1 Broadcast Prevention

Typical IXPs networks forward traffic based on traditional Layer 2 mechanisms. It means broadcast ARP requests and unicast frames with an unknown destination are forwarded to all nodes of the network. Broadcast, flooding and the consequential learning of MAC addresses allow the discovery of nodes connected to the network, easing the insertion of new nodes. However, an excess of broadcast traffic wastes bandwidth and can even lead to network disruption. In IXPs, ARP storms caused by multiple ARP requests to a non-available member or a possible misconfiguration also increase the CPU load of the participants' routers.

The common organization of IXPs, where the MAC addresses of the member are known in advance, and the adoption of SDN capabilities are used in the ENDEAVOUR's platform to eliminate broadcast traffic at the IXP. The "Umbrella" approach encodes the broadcast and destination addresses into multiple labels that provide the path as a series of output ports on every hop until the final destination. For a full description of how "Umbrella" works refer to Deliverable 2.1 (Section 6).

The deployment of "Umbrella" in hardware testbed needs to take into account the capabilities of the underlying hardware switches. One of the core switches of the testbed does not support matching on masked MAC addresses, hence for that reason all virtual MAC addresses created with the encoded path needed to be fully installed on the cores of the Testbed. It decreases the efficiency of the solution that should have just one flow per port on each core switch. Instead every core has to keep the possible virtual addresses for each participant.

Test Description The correct behavior of the broadcast prevention can be observed by the basic usage of the platform. Since "Umbrella" encodes the path of the packets in the network the test simple starts the platform in the testbed and tests if all members can reach it each other. Nonetheless, every broadcast ARP becomes a unicast packet in the edge, so no broadcast addresses should be seen in the core.

Acceptance Criteria For all stages of the development the following individual criteria were used to verify the functional behavior of the use case.

5.1-A No ARP broadcast packet is seen in the core switches. With "Umbrella" every broadcast packet will become a unicast packet at the edges of the IXP.

5.1-B Every ARP packet with a Virtual Next Hop (VNH) as target is sent to the ARP Proxy. If ARP requests to VNH do not reach the ARP proxy the members are not able to exchange traffic in the IXP fabric.

5.1-C All BGP peers connect. "Umbrella" should guarantee that packets are forwarded to the respective routers. Thus checking if BGP works is enough to verify if the all flows from "Umbrella" are installed correctly.

Results from Virtual Testbed The initial implementation of the broadcast prevention use case in the virtual testbed was demonstrated in the Deliverable 4.4. The tests pass every acceptance criteria.

Results from Hardware Testbed The initial results in the Hardware Testbed with the public code on `endeavour/uctrl` folder of the ENDEAVOUR main repository was not satisfactory. Because of the aforementioned issue with the SDN capabilities of one of the core switches, none of the tests worked as expected. After the adaptations to the flows installed by "Umbrella", all the acceptance criteria for the tests were satisfied. A video demonstrating the final functional ability of this use case on the hardware testbed can be found in Deliverable 4.7.

5.2 Access Control

Controlling what traffic is allowed to traverse a network is a crucial, yet cumbersome, operation in today's IXP networks. With hundreds of members sharing the same physical infrastructure, the IXP operators must carefully configure their network devices so prevent any possible source of malicious or unwanted traffic. Yet, traditional IP networks lack the necessary fine-grained technical capabilities required e.g. to block unwanted BGP sessions external to the IXP fabric to be established throughout the IXP network. 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 out packets due to misconfiguration of a member's router.

The deployment on real hardware did not require any change to the implementation of the access control module. The implementation of access control capabilities is part of the Access Control module, which can be found in the `endeavour/acctrl` folder of the ENDEAVOUR main repository. The structure of the Access Control module is built using the same ideas described in Deliverable 3.3 (Section 3.1) for the monitoring module. The access-control rules are installed in a dedicated table that is located right after the Main-Out table. A controller receives access-control rules to be installed in the IXP platform. Those rules are formatted in JSON schema and can easily implement both white- and black-list filtering policies. An example of such rules can be found in:

- `iSDX/test/specs/test1-mh-ac-access_control_flows.cfg`.

We refer the reader to Deliverable 2.3 (Section 3.9) for more details on how the access control rules are structured at the forwarding plane level.

Test Description We now describe how the ENDEAVOUR platform leverages SDN's direct control over packet-processing rules to enable members to express flexible fine-grained policies for access control.

The Access Control use case demonstrator is built upon the DE-CIX physical testbed depicted in Figure 5. The IXP operator wishes to (i) prevent BGP sessions involving non-border routers to be established throughout the IXP networks and (ii) filter out all the OSPF traffic that enters the IXP fabric.

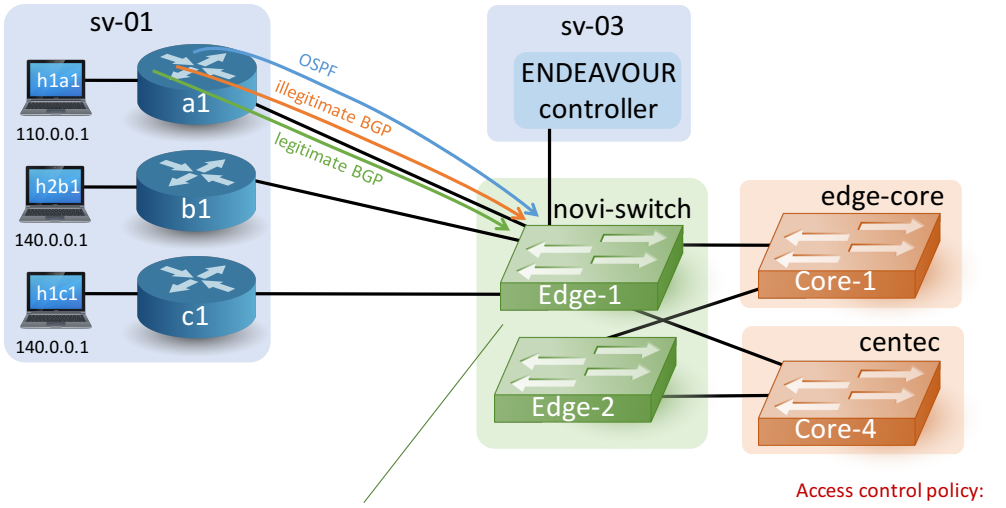
The demonstration evolves as a sequence of two phases:

1. The network is started. BGP sessions among the 3 border routers and the Route Server are established and BGP traffic flows regularly throughout the IXP network.

2. Member `A`'s border router generates both BGP packets towards host `150.0.0.1`, which a destination outside of the IXP fabric and BGP packets towards host `172.0.0.22`, which a destination inside of the IXP fabric.

3. Host `110.0.0.1` generates OSPF packets towards host `140.0.0.1`.

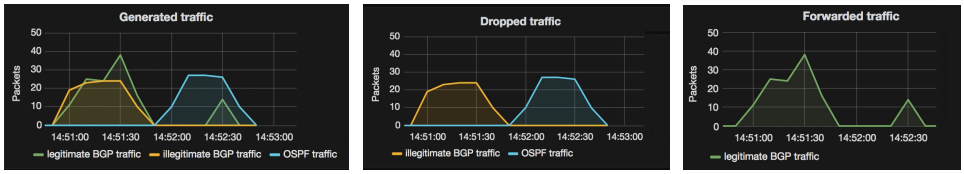
Acceptance Criteria The test scenario that was used at all stages of the development consisted of the following individual criteria and test details:



Access control table

IP src	IP dst	IP proto	TCP dst	GO TO
...	...		...	
172.0.0.0/16	172.0.0.0/16	*	179	load-balancing table
*	*	*	179	drop
*	*	OSPF	*	drop
*	*	*	*	load-balancing table

Figure 5: ENDEAVOUR physical testbed topology for the Access Control use case.



(a) Generated traffic. (b) Dropped traffic. (c) Forwarded traffic.

Figure 6: Demonstration of the Access Control use case. Colors usage: yellow for legitimate BGP traffic, green for non-legitimate BGP traffic, and blue for OSPF traffic.

5.2-A The ENDEAVOUR platform installs the forwarding state into the Access Control tables that reflects the Access Control policies specified by the IXP operator.

5.2-B The flow of BGP traffic generated by member `A` destined to `150.0.0.1` is dropped by the NoviFlow switch.

5.2-C The flow of BGP traffic generated by member `A` destined to `172.0.0.22` is forwarded by the NoviFlow switch.

5.2-D The flow of OSPF traffic generated by `110.0.0.1` destined to `140.0.0.1` is dropped by the NoviFlow switch.

Results from Virtual Testbed Results from the virtual testbed has been extensively described in Deliverable 4.4. All the acceptance criteria were satisfied.

Results from Hardware Testbed All the acceptance criteria were satisfied.* The forwarding state has been installed as described in Deliverable 2.3. We monitored traffic using the Monitoring table of the ENDEAVOUR platform. The last two phases of the test are depicted in Figure 6a, where we use different colored lines to draw the different types of traffic traversing the IXP network: yellow for legitimate BGP traffic (Phase 1), green for illegitimate BGP traffic (Phase 2), and blue for OSPF traffic (Phase 3). Figure 6b and Figure 6c shows what traffic is being dropped and forwarded, respectively. We can observe that legitimate BGP traffic is being correctly forwarded while the illegitimate BGP and OSPF traffic is being dropped. A video of this use case demonstration can be found in Deliverable 4.8.

5.3 Load Balancing

Load balancing is a vital mechanism for IXPs as soon as the infrastructure grows and comprises more than one switch. The common architecture of large IXPs consists of several core switches which transfer traffic between edge switches. The traffic forwarded by the core switches has to be distributed equally. Therefore a load balancing mechanism has to be used. From traditional network devices several implementations are available, but often they have to be built on top of complex setups like routing protocols. The SDN paradigm of ENDEAVOUR opens new possibilities to build a custom load balancing algorithm for IXPs.

A very lightweight approach was chosen for the ENDEAVOUR platform, with an important advantage being its independence, as it is implemented as a core feature of the traffic forwarding and does not require additional network configuration. The basic idea of the solution is to use the IP address as criteria for the decision of the forwarding path. In more detail the source and destination IP address is converted to its binary representation and the least significant bits are used to compute one of the available core switches to transfer the traffic. This approach also provides great flexibility, as the computation from the IP address can be easily adjusted to a different number of core switches or to have a weighted distribution.

During the development process, functional tests have been passed several times in the virtual environment, until the implementation was tested successfully. The test on the hardware succeeded without any adjustments to the original implementation. To determine the performance of the chosen load balancing algorithm, further research with real world data has been conducted, which is described in Section 6.2. The results showed that using the IP address is suitable to be used as criteria for a load balancing algorithm within an IXP and can achieve similar results compared to a more complex algorithm e.g. Equal-Cost Multi-Path Routing (Equal-Cost Multi-Path Routing (ECMP)) [11].

Test Description To conduct tests of the load balancing use case a multi hop topology with at least two core switches has to be available. An even number of flows with an equal traffic volume was transferred over the platform, between different IP addresses. Within the virtual testbed an architecture with four core switches was used, whereas the hardware environment had a reduced number of two core switches to minimize the setup overhead. By monitoring the utilization of all core switches, the correct function of the load balancing algorithm can be verified. A close to uniform distribution of

the traffic between the different core switches was expected.

Acceptance Criteria

5.3-A Utilize multiple available paths to distribute the traffic load among the core switches. The correct function is verified by monitoring the traffic of all core switches. It is expected that all core switches must forward traffic at the same time, therefore traffic is distributed over multiple paths.

5.3-B Uniform distribution of traffic over the utilized paths and core switches. The verification can be done by monitoring the traffic volume over all available core switches. An equal distribution of the traffic volume is expected.

5.3-C The mechanism should be able to spread traffic non-equally among certain links, while taking the available bandwidth of each individual link into account.

Results from Virtual Testbed During development, the load balancing algorithm has been implemented and tested within the the virtual testbed. All features that have been initially drafted within Deliverable 4.2 have been implemented, except a dynamic feature. The idea is to take the load of each individual link into account and distribute the traffic asymmetrically. Nevertheless, the primary objectives for the implementation have been the core features 5.3 and 5.3. With the end of the initial implementation, all main acceptance features have been tested with success. The task of implementing an extended functionality with dynamic abilities was left open but could be implemented on top of the implemented basic features. A demonstration of the use case within the virtual environment is included in Deliverable 4.4.

Results from Hardware Testbed Deploying the load balancing components of ENDEAVOUR in the hardware testbed didn't require any additional changes to the implementation from the virtual setup. Even having a different number of core switches do not require any changes to the algorithm. All tests showed the same positive results from what have been experienced in the virtual environment. A presentation of this use case deployed in the hardware testbed is given by Deliverable 4.7.

6 Feature and Scaling Tests

6.1 Hardware Tests

The ENDEAVOUR project with its deployment of a variety of state-of-the-art SDN switches allow us insights in technical implementation, supported SDN features, and performance. The results were matched with ENDEAVOUR's requirements and lead to the selection or exclusion of hardware switches to be suitable for deployment in our testbed. Unfortunately, due to NDAs between the project partners and the SDN vendors we have to refrain from publishing results of these test.

6.2 Load Balancing Real World Performance

We evaluated the performance of the ENDEAVOUR load balancing mechanism within a real world scenario. Therefore we used a one-day data set from one of the edge switches of a large IXP. The maximum (minimum) observed throughput is 1.2 Tbps at 20:57 PM (0.2 Tbps at 5:03 AM). With this evaluation we where able to show that while IXPs traditionally use the more complex hash-based ECMP to balance the traffic inside the fabric across its edge switches, ENDEAVOUR succeeds with a simpler mechanism. Figure 7 depicts the traffic distribution across the four core switches over time when the ENDEAVOUR load balancing mechanism is in place, achieving significantly better performance, where each of the 4 links constantly receives less than 27% of traffic.

An ideal ECMP scheme would obtain an almost perfect balancing at the cost of a higher complexity for the computation, algorithms or basic setup.

While between 7 AM and 9 AM, our solution provides an almost perfect traffic load balancing, from 0 AM to 6AM we observed a slight unbalance. To gain further insights, we analyzed the size of the flows and their total traffic share, as illustrated in Figure 8. The two measurements `t1` and `t2` spread over 1 hour each, with `t1` starting at 1 AM and `t2` at 7 AM. We define a flow as the traffic exchanged between a pair of source and destination IPs. Interestingly, a larger fraction of smaller traffic flows is forwarded during `t2`. This is consistent with previous studies on data centers, where static load balancing techniques gain in performance by increasing the fraction of smaller flows [1]. The ENDEAVOUR load balancing mechanism is hence appropriate, as IXP traffic is typically characterized by smaller flows due to the traffic requested by eyeball and ISP networks [4].

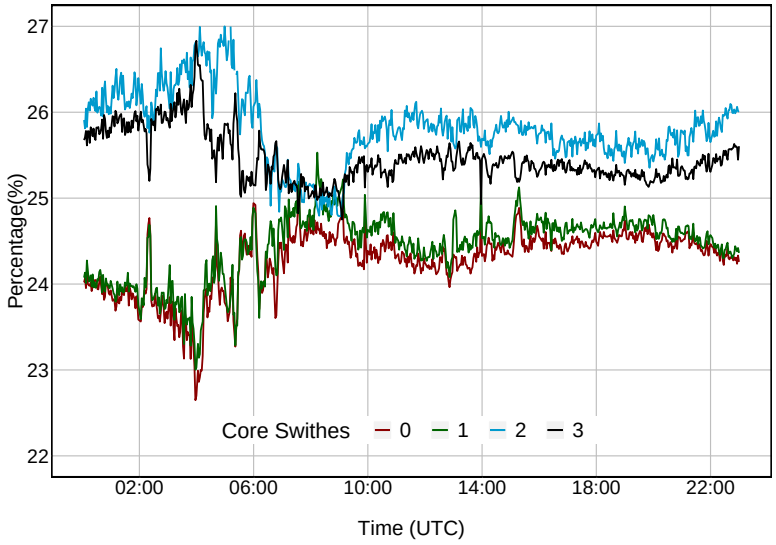


Figure 7: Load Balancing real world performance.

6.3 Blackholing Flow Rule Deployment Time

To evaluate the blackholing capabilities of the ENDEAVOUR platform further, we used the BGP blackholing updates of an real world IXP [7] over the course of one day. These announcements and withdrawals are then replayed by using the ENDEAVOUR blackholing API. Figure 9 depicts the installation time of the new updates and reports the total number of rules installed in the fabric. Updates are performed in blocks of 50. We measured the time from calling the API to the application of the last rule on the data plane.

Figure 9 shows how the first block of updates is completed within 7 and 9 seconds. When issuing 1300 updates in consecutive blocks, times raise above 10 sec. The number of rules installed in the fabric scales to 2122 after 2000 replayed flow rule updates: the time until rules' activation grows proportionally.

From evaluation we learn that the ENDEAVOUR blackholing API is able to provide an blackholing service with SDN. Furthermore, we where able to replay a realistic sample day of blackholing. We learn that the time for blackholing rules to take effect is proportional to the number of installed rules in the fabric.

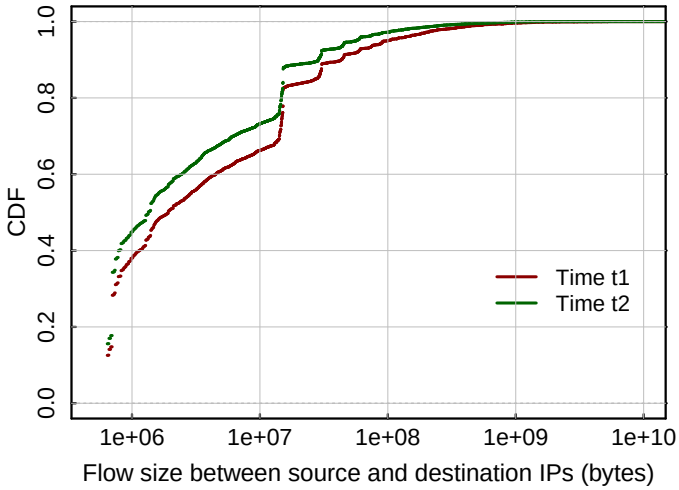


Figure 8: Distribution of flow sizes within the performance measurement.

7 Summary

In this report we evaluate the correct function of the ENDEAVOUR platform and examine critical performance aspects. Within Section 3, we provide an overview of the results of the tests that have been conducted for all use cases. The results show that all test criteria are passed and all use cases are implemented successfully. The methodology of the anomaly detection use case relies on hardware support. Therefore, there was no test within a virtual environment for this use case. All other use cases were initially implemented in a virtual testbed, that was executed on servers in the DE-CIX testbed. Thus, each use case could be implemented and tested in parallel, which accelerated the process of the implementation significantly. The implementation of all features of the ENDEAVOUR platform have been accompanied with regular checks on their functional correctness and completeness following the test design described in 3. Different SDN switches have been acquired, tested and finally utilized to build an IXP architecture, forming the hardware testbed for the ENDEVOUR platform. Subsequently, all use cases have been transitioned to the hardware testbed, which all partners within the ENDEAVOUR consortium shared. Fortunately our careful implementation and extensive testing within the virtual environment turned out to be successful and valuable for the overall project proceedings. All use cases have been transferred to the hardware testbed, with only minor adjustments that were necessary to have all features finally pass the pre-defined acceptance

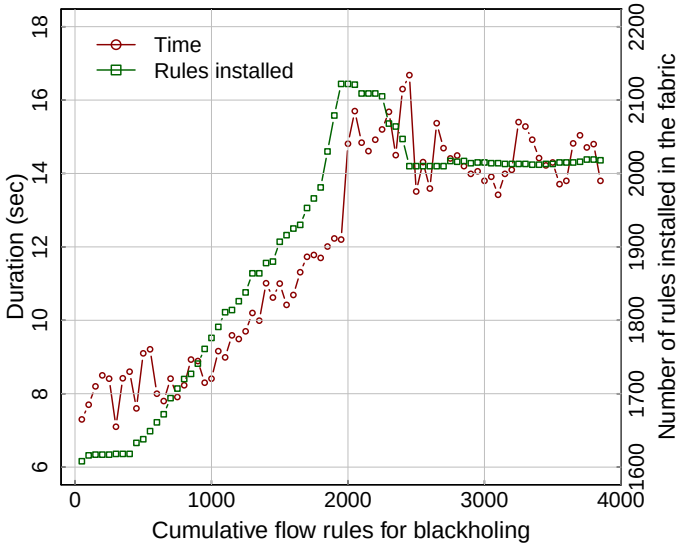


Figure 9: Blackholing rule deployment time.

criteria. Beyond the regular tests, additional examinations about a variety of performance and scaling aspects have been conducted. Thus, we showed an excellent performance of the load balancing algorithm of ENDEAVOUR within a real world scenario. We ensured that the blackholing feature is able to work within a realistic environment and provided details on performance parameters of the platform. The test design used with ENDEAVOUR provided the best support during the development process. The additional performance and scalability tests have proven not only that ENDEAVOUR can be deployed in real world scenarios, but also provide valuable insights for the research community [2].

8 Acronyms

PoW Proof of Work

SDN Software Defined Networking

BGP Border Gateway Protocol

ISP Internet Service Provider

IXP Internet eXchange Point

AS Autonomous System

IP Internet Protocol

OSPF Open Shortest Path First

DE-CIX German Commercial Internet Exchange

VNH Virtual Next Hop

DDoS Distributed Denial of Service

TE Traffic Engineering

VM Virtual Machine

TCP Transport Control Protocol

UDP User Datagram Protocol

ARP Address Resolution Protocol

ECMP Equal-Cost Multi-Path Routing

API Application Programming Interface

MAC Media Access Control

iSDX industrial Software-Defined-Exchange

HTTP HyperText Transfer Protocol

JSON JavaScript Object Notation

OSNT Open Source Network Tester

ORUNADA Online and Real-time Unsupervised Network Anomaly Detection Algorithm

NDA Non-disclosure agreement

References

[1] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *NSDI*. USENIX, 2010.

[2] Gianni Antichi, Ignacio Castro, Marco Chiesa, Eder L. Fernandes, Remy Lapeyrade, Daniel Kopp, Jong Hun Han, Marc Bruyere, Christoph Dietzel, Mitchell Gusat, Andrew W. Moore, Philippe Owezarski, Steve Uhlig, and Marco Canini. ENDEAVOUR: A Scalable SDN Architecture for Real-World IXPs. *IEEE Journal on Selected Areas in Communications*, 35(11), Nov 2017.

[3] Sascha Bleidner, Christoph Dietzel, and Marc Bruyere. ENDEAVOUR Deliverable 4.2: Design of Use Cases for Operators of IXPs, 2016.

[4] Nikolaos Chatzis, Georgios Smaragdakis, Jan Böttger, Thomas Krenc, and Anja Feldmann. On the Benefits of Using a Large IXP As an Internet Vantage Point. In *IMC*. ACM, 2013.

[5] Christoph Dietzel, Gianni Antichi, Ignacio Castro, Eder L. Fernandes, Marco Chiesa, and Daniel Kopp. SDN-enabled Traffic Engineering and Advanced Blackholing at IXPs. In *SOSR*. ACM, 2017.

[6] Christoph Dietzel, Sascha Bleidner, G Kathareios, P Owezarski, S Abdellatif, M Chiesa, M Canini, and Antichi. ENDEAVOUR Deliverable 4.3: Design of Use Cases for Members of IXPs, 2016.

[7] Christoph Dietzel, Anja Feldmann, and Thomas King. Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild. In *PAM*, 2016.

[8] J. Dromard, G. Roudière, and P. Owezarski. Orunada, an online and real-time unsupervised network anomaly detector. *IEEE Transaction on Network and System Management (TNSM)*, 2016.

[9] Fernandes, Boettger, Deng, and Castro. D3.4 Deployment of the prototype of the monitoring platform. In *Deliverable of the H2020 ENDEAVOUR project*, Dec 2017.

[10] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger. Inferring BGP Blackholing Activity in the Internet. In *ACM Internet Measurements Conference (IMC)*, 2017.

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