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	4.8
Document title	Final Implementation of Selected Use Cases for IXP Members
Document version	0.6
Editor in Chief	Kopp, DE-CIX
Authors	Kopp, Dietzel, Chiesa, Lapeyrade, Owezarski
Date	20/12/2017
Reviewer	Owezarski, CNRS
Date of Review	20/12/2017
Status	*Public*

Revision History

Date	Version	Description	Author
27/11/17	0.1	First structure	Kopp
11/12/17	0.2	Inbound/Outbound TE added	Chiesa
11/12/17	0.3	Advanced Blackholing added	Kopp
15/12/17	0.4	Hardware Testbed added	Kopp
18/12/17	0.5	General text added	Dietzel
19/12/17	0.6	Review	Kopp, Dietzel
20/12/17	0.7	Final Review	Owezarski

Executive Summary

Over the course of the ENDEAVOUR project the consortium developed a wide range of use cases as potential candidates to be implemented within the ENDEAVOUR prototype. After consolidating the most compelling use cases we implemented them into the virtual ENDEAVOUR platform. Consecutively, we transitioned the implementation from the virtual testbed to real hardware switching fabric testbed in the DE-CIX data center. To this end, the present deliverable showcases the final implementation of use cases of the ENDEAVOUR platform for Internet eXchange Point (IXP) members. Each use case is demonstrated in a video. In addition, Deliverable 4.7 discusses the relevant use cases for IXP operators. In combination, these two deliverables reflect the final state of the ENDEAVOUR platform prototype, extensively tested and reported in Deliverable 4.6. To make the demonstrations easily accessible we present technical background necessary to understand the workflow of each use case.

Contents

1 Introduction

In this report, we present the final implementation of the ENDEAVOUR use cases for Internet eXchange Point (IXP) members. This includes demo videos for all use cases and a brief description of the implementation of all demonstrators. We first provide a pointer to the implementation of ENDEAVOUR in Section 2 and describe the hardware testbed in Section 3. Then we present the demonstrated use cases (Section 4) together with a short technical description. The final Section 5 summarizes the report.

Note, that this document is not self-contained, but accompanies the demonstrator videos[1] and the code[2]. Details on the implementation of the ENDEAVOUR framework can be found in Deliverable 2.3 [1] and a description of the deployment in Deliverable 2.4 [2]. While the monitoring architecture is discussed in Deliverable 3.3 [5] the details on the deployment of the Software Defined Networking (SDN) prototype can be found in Deliverable 3.4 [6]. For a comprehensive description of the use cases for IXP members we refer to Deliverable 4.3 [3] and to Deliverable 4.5 [4] for the demonstrator in a virtual environment.

2 Software Implementation of ENDEAVOUR

The ENDEAVOUR consortium maintains its code on the GitHub platform. We agreed on only having deployable code in the master branch. The ENDEAVOUR GitHub repository provides the code to build the ENDEAVOUR platform. Nevertheless, the final implementation is hardware dependent and optimized for the current setup deployed in the DE-CIX data center.

3 Hardware Testbed

The hardware testbed represents a real-world setting of an IXP by consisting of hardware, in contrast to software, switches. It comprises three switches from different vendors and four servers and is deployed at one of the DE-CIX data centers. The final network configuration of the testbed is depicted in Figure 1, which consists of an edge-core IXP topology with two edge switches interconnected to both core switches.

The four servers are `sv-01`, `sv-02`, `sv-03`, and `sv-08`. Participant network devices are executed on `sv-01`, `sv-02`, and `sv-08` through `docker` con-

[1] `https://www.youtube.com/playlist?list=PLl6z513p1ClEJc0PB8IlMURV_qSg1pZyD`

[2] GitHub - `https://github.com/h2020-endeavour/endeavour`

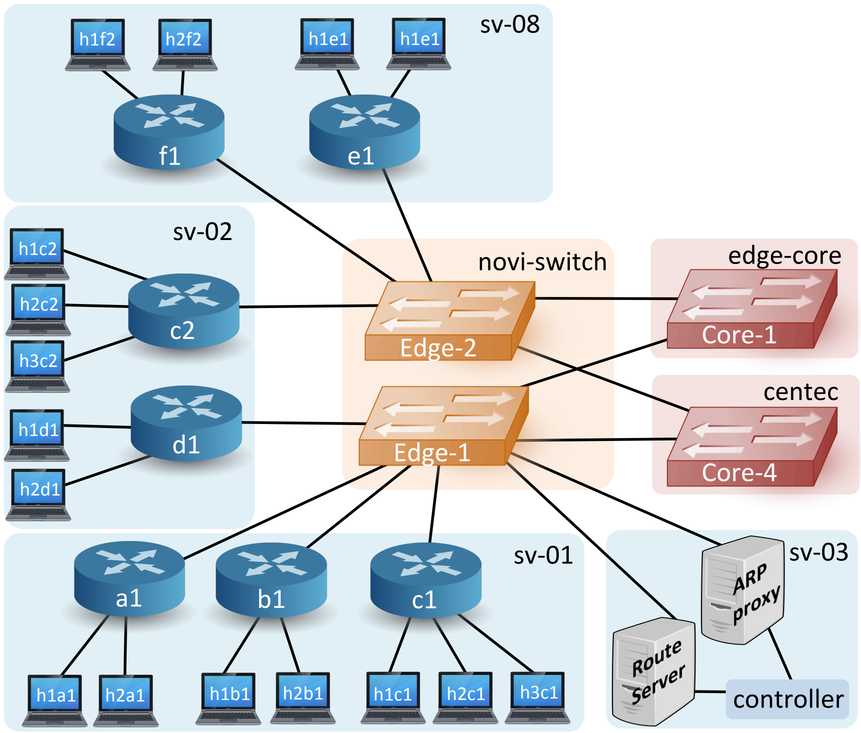


Figure 1: Testbed environment

tainers. The ENDEAVOUR controller, the route server, and the Address Resolution Protocol (ARP) proxy are executed on `sv-03`. The hardware switches used in the testbed are: `novi-switch`, `edge-core`, and `centec`. We used the `centec` and `edge-core` switches as the core switches of the IXP fabric.

To run the ENDEAVOUR platform within the hardware testbed at the DE-CIX data center complete the following steps:

1. The docker containers on `sv-01`, `sv-02`, and `sv-08` have to be started by the following command:

   ```
   $ cd docker && sudo ./scripts/sdx_start_docker \
   AS_config/bgp-tst/quagga/ AS_config/test_containers
   ```

2. To access the ENDEAVOUR platform components, go to `sv-03` and start the prepared tmux sessions with:

```
endeavour@sv-03:~$ ./setup-tmux-for-test-te-acctrl.sh
```

3. After starting the docker container and having the tmux sessions for the ENDEAVOUR controller ready, the actual ENDEAVOUR controller can be started. This can be done from within the `endeavour` session in tmux on `sv-03` and issuing the following command:

   ```
   endeavour@sv-03:~$ tmux attach -t endeavour
   endeavour@sv-03:~$ cd /endeavour/
   endeavour@sv-03:$ ./launch.sh --stats --acctrl test-te 3
   ```

4. Form the last step on the ENDEAVOUR platform is ready and running. As an optional step, tmux sessions on `sv-01`, `sv-02`, and `sv-08` can me started to support the demonstration of the use cases:

   ```
   ubuntu@sv-01:~$ cd ./docker/scripts/tmux/
   ubuntu@sv-01:$ ./setup-tmux-for-test-te-acc.sh
   ubuntu@sv-02:~$ cd ./docker/scripts/tmux/
   ubuntu@sv-02:$ ./setup-tmux-test-te.sh
   ubuntu@sv-08:~$ cd ./docker/scripts/tmux/
   ubuntu@sv-08:$ ./setup-tmux.sh
   ```

This completes the start of the ENDEAVOUR platform within the hardware testbed. The following sections demonstrate each use case, executed within the hardware testbed. All use case demonstrations rely on the basic setup of the ENDEAVOUR components as described in this section.

4 Demonstration of Member Use Cases

4.1 Inbound/Outbound TE

TE The Inbound/Outbound use case demonstrator is built upon the hardware testbed at DE-CIX using simplified IXP scenario depicted in Figure 2. Member `A` wishes to send HTTP traffic towards member `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 six 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 Border Gateway Protocol (BGP) announcements and creates the necessary Virtual Internet Protocol (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 one-minute 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 one-minute 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.

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

- 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.
- The first flow of traffic generated by member A destined to port 80 is correctly received by member B.
- 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.
- 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.
- The flow of traffic generated by member A destined to port 4321 is correctly received by member C through port C1.

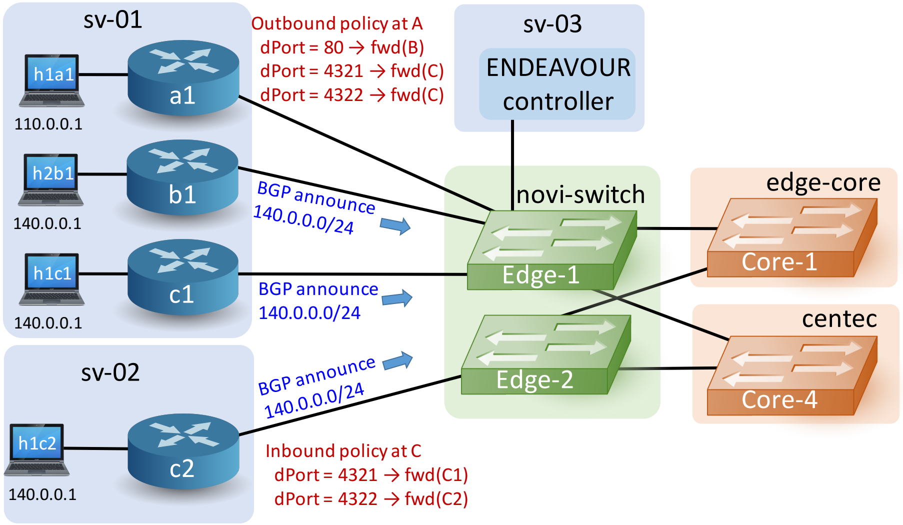


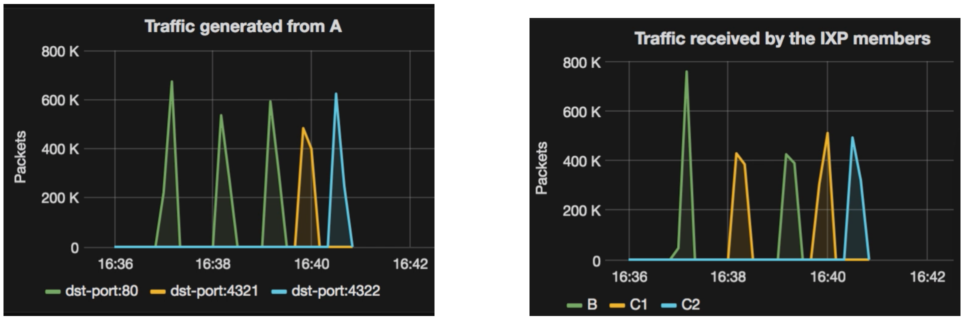
Figure 2: ENDEAVOUR hardware testbed topology for the Inbound/ Outbound TE use case.

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

All the above mentioned acceptance criteria were satisfied. The forwarding state has been installed exactly as described in D2.3 (more details in the demonstrator video provided below). We monitored traffic using the Monitoring table of the ENDEAVOUR platform. The five phases related to the last five acceptance criteria are depicted in Figure 3a, where we use different colored lines to draw the different types of traffic entering the IXP network from member `A`. Figure 3b shows how traffic is being received by members `B` and `C`. We can observe that HTTP traffic 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 of this use case demonstration can be found at:

- `https://youtu.be/Tjq3Bi_8s5E`



(a) Traffic originated from `A1`.

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

Figure 3: Demonstration of the Inbound/Outbound TE use case.

4.2 Advanced Blackholing

The advanced blackholing feature of the EANDEAVOUR platform is utilized to block malicious traffic and prevent Distributed Denial of Service (DDoS) attacks from congesting a member's connection at an IXP. This use case is demonstrated in the testbed by initiating traffic flows between a multitude of hosts between autonomous systems of the IXP. Among these flows there are specific flows, which mimic a DDoS attack. Target for the attack is one of the hosts in the testbed. The autonomous system under attack installs a blackholing rule by using the ENDEAVOUR advanced blackholing Application Programming Interface (API). It defines a detailed rule with source-, destination IP address and port matching only the malicious traffic. The attack traffic is blocked and the rest of the traffic directed towards the victim host remains unaffected. As soon as the attack has stopped, the blackholing rule can be deleted. A simplified visualization of the hardware testbed's topology used for demonstrating this use case is given in Figure 4.

1. Before the test case starts we ensure the testbed has converged. When all network layer reachability information is exchanged the ENDEAVOUR platform is up and running.

2. Legitimate traffic is generated from hosts `110.0.0.1, 130.0.0.1, 140.0.0.1, 190.0.0.1, 210.0.0.1 and 220.0.0.1` to destination `170.0.0.1` on port 80.

3. Simulated attack traffic flows from `200.0.0.1` to `170.0.0.1` on port 53.

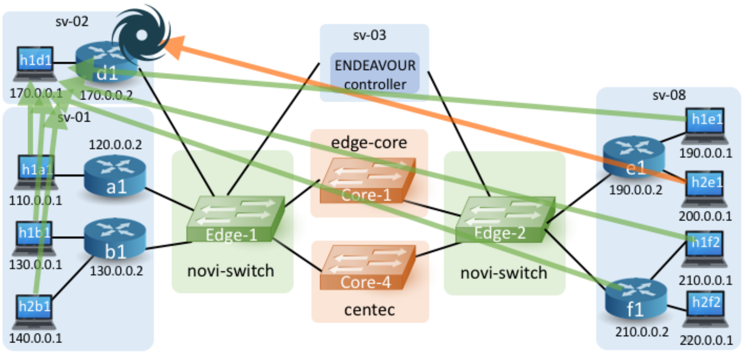


Figure 4: Blackholing topology

4. Attack traffic disrupts normal flow of traffic towards `170.0.0.1`. In the testbed attack traffic is generated the same way as ordinary traffic, but on a different port.

5. Now the network under attack installs a blackholing rule to block the attack traffic.

6. Traffic from `200.0.0.1` to `170.0.0.1` on port 53 is blackholed.

7. Since the attack traffic is blocked, all other traffic flows are normal.

8. Finally the blackholing rule is removed, the traffic flow between `200.0.0.1` and `170.0.0.1` on port 53 is resumed (attack traffic).

The monitoring feature of ENDEAVOUR can be used to observe effectiveness and correct behavior.

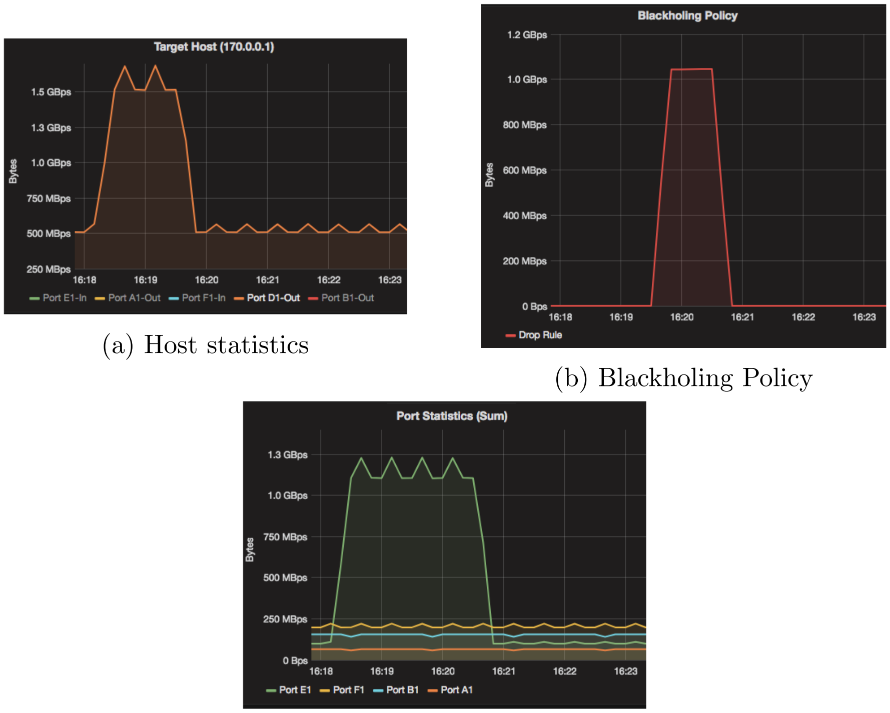
1. Check the traffic flow in target tost `170.0.0.1` see Figure 5a.

2. Figure 5b describes the port statistics of different routers involved in the test case.

3. Once the blackholing rule is installed the traffic in target host is stopped, this can be visualized in 5a and 5c.

The demonstration of the blackholing use case deployed in the hardware testbed can be found here:

- `https://youtu.be/hmDIlJFJwWc`



(a) Host statistics

(b) Blackholing Policy

(c) Port statistics

Figure 5: Demonstration of Blackholing use case

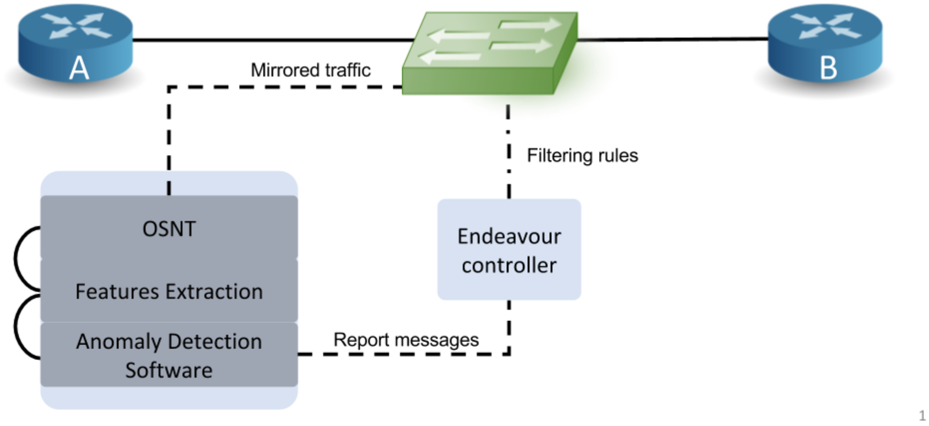


Figure 6: Topology for the Anomaly detection traffic use case

4.3 Traffic Anomaly Detection

For this demonstration, we will show how the ENDEAVOUR platform can enforce filtering policies, with the help of the traffic anomaly detection tool. To showcase this claim, we perform the demonstration on the topology presented in Figure 6. We consider two routers, named router `A` and router `B`, which are connected to an IXP. This IXP is running a SDN switch, controlled by the ENDEAVOUR platform. The owner of router `B` asked to probe its received traffic in order to block potential anomalies. As a consequence, the outbound traffic for this router will be mirrored to a dedicated port.

To visualize this attack, we show a tcpdump filtering the upcoming attack.

The demonstration is divided into the following steps:

1. The participants are peering their IP addresses at the beginning of the video. It first exhibits some BGP traffic exchanged in router `B` traffic visualized thanks to tcpdump.

2. Router `A` sends some traffic to router `B`. The IXP forwards this traffic correctly.

3. Router `A` begins to send malicious traffic.

4. The anomaly detection platform detects this attack, sending a report message to the ENDEAVOUR platform.

5. The ENDEAVOUR platform applies a filtering rule based on informations sent by the anomaly detection software. It then stops forwarding this attack to router B.

The demonstration of the blackholing use case deployed in the hardware testbed can be found here:

- https://youtu.be/c3CUkM1uF0E

5 Summary

In this report, we describe the demonstrator for the final implementations of ENDEAVOUR use cases for IXP members. Thereby, we aim to provide in addition to the demo videos an outline of the setup of the environment, then we present the implemented use cases together with technical details. Finally, we discuss the workflow of the demonstrator to make our results even more accessible.

6 Acronyms

SDN Software Defined Networking

BGP Border Gateway Protocol

IXP Internet eXchange Point

IP Internet Protocol

DDoS Distributed Denial of Service

TE Traffic Engineering

ARP Address Resolution Protocol

API Application Programming Interface

MAC Media Access Control

HTTP Hyper Text Transport Protocol

References

[1] Canini, Chiesa, Dietzel, and Fernandes. D2.3 Implementation of the SDN Architecture. In *Deliverable of the H2020 ENDEAVOUR project*, Dec 2016.

[2] Chiesa. D2.4 Deployment of the Prototype SDN Architecture. In *Deliverable of the H2020 ENDEAVOUR project*, Dec 2017.

[3] Dietzel, Bleidner, Kathareios, Chiesa, Castro, Abdellatif, Antichi, Bruyere, Fernandes, and Owezarski. D4.3 Design of Use Cases for Members of IXPs. In *Deliverable of the H2020 ENDEAVOUR project*, Jan 2016.

[4] Dietzel, Kopp, Abt, Chiesa, and Owezarski. D4.5 Implementation of the Selected Use Cases for the IXP Members. In *Deliverable of the H2020 ENDEAVOUR project*, Jan 2017.

[5] Fernandes, Boettger, Antichi, Lapeyrade, and Owezarski. D3.3 Implementation of the Monitoring Platform. In *Deliverable of the H2020 ENDEAVOUR project*, Jan 2017.

[6] Fernandes, Boettger, Deng, and Castro. D3.4 Implementation of the Monitoring Platform. In *Deliverable of the H2020 ENDEAVOUR project*, Dec 2017.