30-10-2019

Deliverable D6.3
White Box Evaluation

Deliverable 6.3

Contractual Date:	31-10-2019
Actual Date:	30-10-2019
Grant Agreement No.:	856726
Work Package	WP6
Task Item:	Task 1
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	RENATER
Document ID:	GN4-3-19-23B128
Authors:	Xavier Jeannin (RENATER), Mauro Campanella (GARR), Frederic Loui (RENATER), Edin Salguero (RENATER), Maxime Wisslé (RENATER), Christos Argyropoulos (GRNET), Jani Myyry (FUNET), Ivana Golub (PSNC), Tomasz Szewczyk (PSNC), Damian Parniewicz (PSNC) , Bojan Jakovljevic (AMRES), Pavel Benacek (CESnet), Marco Savi (GARR), Susanne Naegele Jackson (FAU/DFN), Tim Chown (Jisc)

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).

Abstract

This deliverable reviews whether new types of white box devices can be used by the research and education community and for which use cases. The ability to program the data plane thanks to a high abstract level language (P4) opens the door to new applications for research and education. Two use cases are presented here: In-band Network Telemetry and Distributed Denial of Service attack mitigation. The Router for Academia, Research and Education project (RARE) investigates also if it is possible to use open source Network Operating System (NOS).

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This deliverable reviews whether new, emerging types of white box hardware may be used as switches or routers by the research and education community and for which use cases.

A white box is a switch/router manufactured from commodity components that allows different Network Operating Systems (NOSs) to be run on the same piece of commodity hardware, decoupling the NOS software from the hardware. (Optical white boxes are out of scope of this report.) White boxes, first deployed widely in data centres, offer an impressive forwarding capacity for a very low price. Although current NOS platforms do not provide all the features required by NRENs, the white box approach has the advantage of improving an NREN's level of independence from router vendors and could thus change the way NRENs manage their network deployments. The white box chipset forwarding characteristics (forwarding capacity, internal memory, size of buffers) determine the scenarios in which it can be used (e.g. IX switch, data centre, CPE, P/LSR, etc.).

By exploring several use cases, the GN4-3 *Network Technologies and Services Development* Work Package, Network Technology Evolution task (WP6 T1) presents in this document its work to date in exploring how white boxes can be used for CPE and Internet eXchange point switch use cases. The work on the DC fabric use case is also promising, even if the technical analysis is not finished in this last case. However, the business decision to go into production is not only based on technical considerations and total cost of ownership but also on internal organisational constraints (such as team workload, capacity to hire staff, strategic plan, etc.). It should also be noted that for use cases that require more routing features, like Label Edge Router / Provider Edge (LER/PE), the currently available NOS currently could have limitations.

Thanks to data plane programming (DPP), advanced network features can be programmed for NREN needs. DDoS mitigation algorithms have been implemented on a virtual P4 environment and the implementation on P4-capable hardware is ongoing. In-band Network Telemetry (INT) with P4 allows very accurate network monitoring, debugging in novel ways and can significantly improve network management, using just a few nodes supporting INT.

The Router for Academia, Research and Education (RARE) project aims to demonstrate that an open source control plane on a white box can be used as a router. Continuing the work completed to date on the development of open source data plane routing features and the integration of an open source NOS (for instance FreeRtr) on the P4 data plane, RARE is now working on CPE and P implementations, but there is no theoretical limitation for other use cases.

1 Introduction

The networking industry landscape is evolving fast and the market trend is now directed towards data centre and cloud-based services. The strategy of new players who want to enter this market is to propose not only lower prices and a higher port density ratio, but also to decouple the network operating system (NOS) from the hardware in order to remove their potential customers' dependency on the traditional monolithic vendor router/switch market. This poses the question whether, at the network level, the GÉANT community is in the same situation now as when Linux appeared in the UNIX world. Is white box a real opportunity for NRENs and research and education (R&E) networks?

The second significant evolution is white box programmability, thanks to recent advancements in data plane programmability and new chip implementations (e.g., Barefoot Tofino). P4, a high-level language for data plane programming has been developed to make the data plane programmable, capitalising on the OpenFlow experience.

Data plane programming (DPP) allows line-rate packet processing. Powerful algorithms can be compiled and executed directly in the data plane. This opens the door to the design and development of many potential new features or new improvements. Of these, the GN4-3 *Network Technology Evolution* task in the *Network Technologies and Services Development* Work Package (WP6 T1) selected new network monitoring solutions, In-band Network Telemetry (INT), and a new security solution for DDoS detection and mitigation to demonstrate how DPP might benefit NRENs.

The ability to integrate different pieces of software (control plane, data plane and intercommunication between these two components) is an opportunity to run an open source or commercial NOS over white box hardware. The Router for Academia, Research and Education (RARE) project will investigate, as a first stage, the feasibility to integrate an open-source network control plane that provides a complete feature set compliant to research and education ecosystem requirements, and to connect this control plane to a P4 data plane.

This document reports on the evaluation of white box and data plane programming use in the NREN context. Section 2 details the investigation and the results regarding white box usage (white box for research and education). Section 3 presents the data plane programmability (DPP) work and section 4 reports the work of the RARE team. These sections are then followed by a general conclusion in Section 5.

2 White Box

While there are multiple definitions for white box, within the scope of the work of WP6 T1, it is generally considered to be a switch/router that is manufactured from commodity components and on which different open source or commercial Network Operating Systems (NOSs) can be installed. WP6 T1 is studying white boxes in the NREN context, rather than simply focusing on the context of data centre use cases where white boxes are often presented on the Internet. Optical white boxes are also out of scope of this task.

The business model for proprietary hardware forces anyone who is buying a router to acquire a package comprising certain hardware, a proprietary NOS, the associated hardware maintenance and NOS maintenance. In the case of a white box, the business model allows customers to choose to buy hardware with its maintenance from a hardware supplier and then either buy a commercial NOS or install an open source NOS with maintenance from a software supplier. This provides independence from the hardware (the customer can change the hardware vendor and keep the software) and independence from the NOS (the customer can change the NOS and keep the hardware). To evaluate the potential interest in white boxes within research and education, the Task is analysing the white boxes available on the market, focussing on their applicability and usability in the NREN context.

The Open Compute Project [OCP] specifies an open source initiative called the Open Network Install Environment [ONIE], which defines an open "install environment" for the installation of different NOSs on bare metal switches. Some white boxes can also be provided with a Linux system that allows the installation of a NOS (see Figure 2.1).

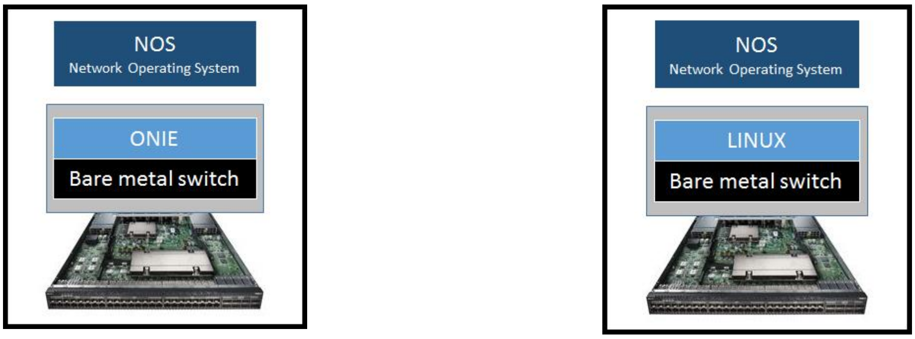


Figure 2.1: White box architecture

Since white boxes were conceived and first deployed in the context of data centres, where high-speed local interconnects are required, white box designers focus on significant (several 100Gbps) forwarding capacity, but with features that aim to address the data centre market (a small number of routes, lots of layer 2 features etc.). A marked difference with regards to traditional network provider chassis routers is that a white box does not exist as a network chassis and does not provide several 'route engine cards' (several CPUs). Some white boxes are equipped with exactly the same chipsets used by traditional vendors [Merchant Chips]. The price of this forwarding capacity is very competitive for this type of hardware. There are different switch designs for different types of usage: data centre, LAN, campus network or network backbone. The first white boxes were designed for data centre (DC) deployment, which implies a very short Round-Trip delay Time (RTT). Such machines were designed to handle microbursts that could occur in a DC (for instance TCP Incast traffic). This led to a design with a relatively short buffer. As white boxes are now deployed more commonly, in a wider range of use cases, white box designers are now targeting new markets and a white box equipped with a large buffer forwarding chip is emerging (Jericho 4GBytes) [Packet buffers]. Section 2.1.2 discusses the importance of the switch buffer size.

Recently, server suppliers have put a hardened X86 server on the market specially designed to become a small router (switch form factor, no graphic card, hardware hardened, designed to be used without cooling, etc.) [X86 router]. As different NOSs can be installed on this machine, it can also be considered a white box. NRENs who express their interest in trying white boxing want to be able to test them with a minimal risk, i.e. at the edge of their network, for instance with a site router use case, Customer Premises Equipment (CPE). As most white boxes previously available on the market are very powerful in terms of forwarding (several 100Gbps ports), they are not really adapted to fit use cases that do not require such capacity. In this context, this new type of machine (the X86 server) can be appropriate for these types of use cases, such as a CPE. Figure 2.2 presents an example of a CPE design and its architecture.

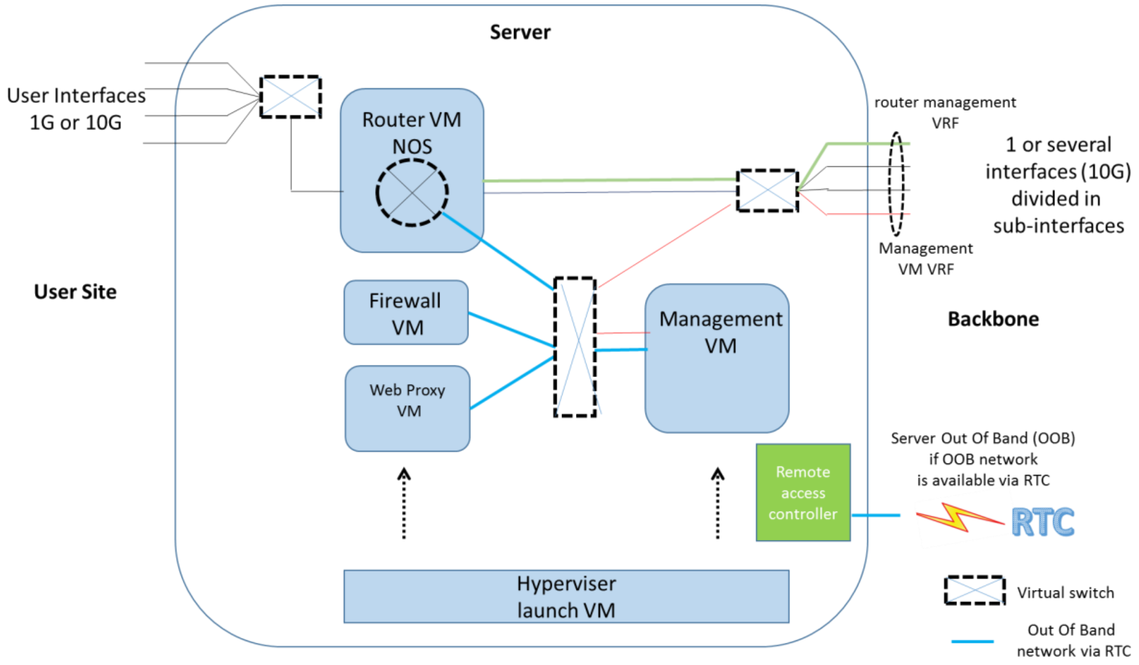


Figure 2.2: Example of CPE design over an X86 server

As shown in Figure 2.2, the router is a virtual machine managed by a hypervisor. It is possible to deploy other virtual machines, implementing different network virtual functions (firewall, WebProxy) that are interconnected through a virtual switch. As this physical server is not equipped with dedicated forwarding, the forwarding capacity is limited and would decrease according to the number of implemented network virtual functions and the activated feature (for example, deep inspection).

2.1 White Box for Research and Education

The first step in evaluating white boxes for R&E is to ascertain which devices are available on the market now or will become available during the project. To make this assessment, several selected NREN use cases are evaluated, with the investigation covering the aspects required for production. The aspects to consider for deployment of such white boxes include routing, management (monitoring, authentication, maintenance model, etc.), security and the license model. The cost is an issue for each NREN to consider internally when they make their business decision whether to deploy white boxes in production. Other points that NRENs must consider before adopting white boxes are their capacity to manage a new NOS and whether the platforms have the necessary maintenance in place. The management of white boxes can differ from that of a traditional switch or router due to the maintenance model. A white box might be maintained by two different companies, one looking after the hardware and another one after the NOS.

2.1.1 NREN Requirements and Concerns

During the White Boxing workshop in Stockholm on 04 April 2019 (for which 40 people, including people from 15 NRENs, registered) [Workshop], WP6 T1 conducted a survey on NREN interest, potential use cases and potential concerns. As Figure 2.3 shows, they indicated three use cases they started with: CPE, cloud fabric and 'big science' projects (Large Hadron Collider (LHC), High Performance Computing (HPC), Large Synoptic Survey Telescope (LSST), etc.). Their concerns were related to support, the quality of software, and reliability.

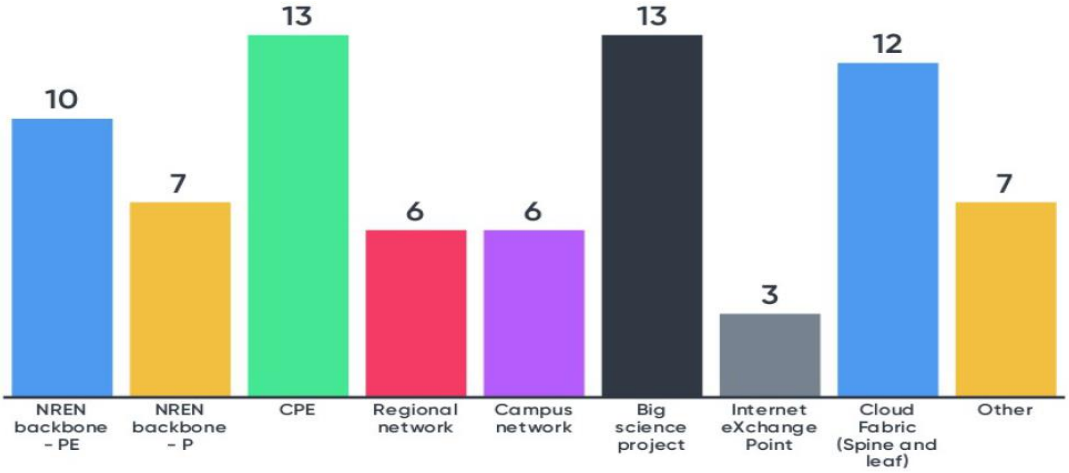


Figure 2.3: Use cases selected by European NRENs for white box usage

The NRENs identified the following points as critical, in order of importance: support, software quality, availability of features, stability and reliability.

These critical points and expressed concerns are taken into consideration during the work of WP6 T1, as presented in the following sections.

2.1.2 Buffer Size

NREN engineers expressed their concerns regarding the buffer size in white boxes compared to 'traditional' routers, and its potential impact on traffic in cases of congestion and/or QoS usage. To address this point, WP6 T1 studied white box behaviour in cases of congestion. The first commercial white box deployments in data centres did not require large buffers. However, now, new white boxes are available that are equipped with large buffers and new chipsets (for instance Jericho).

Typical network traffic can at the same time contain elephant and mice flows, and the management of such flows can additionally be impacted with QoS mechanisms in place. This helps in the creation of microbursts, for which not a single definition could be found, and it is very difficult to obtain information from router manufacturers. Even though microbursts can be seen in a network, the question remains when and where they occur, and which applications are sensitive to delay variation.

In network devices, the buffers function as microburst absorbers. Buffers delay the traffic a little so that the microburst can be absorbed by the overloaded interface. If one wants to manage oversubscription with QoS mechanisms then a large buffer is needed.

Researchers from Stanford and the University of Toronto tried to address this by conducting an experiment on the Level 3 commercial backbone with OC-48 links, buffer size = 60 MB / 190 msec. or 125,000 500B packets with no active queue management [Buffers]. The links were set to the experimental values 1, 2.5, 5 or 10 msec. buffer. No drop was seen with the 5, 10, and 190 msec. buffers for the entire duration. Packet loss in the range of 0.02% to 0.09% was seen with 2.5 msec. of buffering and correlated to the link utilisation. There was a relatively large increase in packet loss with 1 msec. of buffering, but link utilisation was still maintained. Most of the loss occurred when the link utilisation was above 90% for a 30 second average. The packet drop level for the 1 msec. buffer was still below 0.2%.

In the data centre, TCP Incast traffic is generated by application requests (Hadoop, Map Reduce, HDFS for instance) to several nodes that answer in general with very short-lived flows but simultaneously generating microbursts. Researchers at the University of California at San Diego recently performed an in-depth analysis of traffic at Facebook. Servers were 10Gbps attached, their utilisation was under 10% (1% most of the time) and the data on buffer utilisation was collected at 10 µs intervals for links to web servers and cache nodes. The conclusion was that on the ToR switches (Facebook Wedge with Broadcom's Trident II ASIC, which has 12 MB of shared buffers), over two-thirds of the available shared buffers were constantly in use during each measured interval [Roy_et_al].

Based on a study reviewed by WP6 T1 [Packet_buffers], the following table summarises the applications that could be impacted by packet loss and delay variation:

Application	
High-Frequency Trading	Device latency must be eliminated and buffering minimised.
Gaming	Usage of buffers could be beneficial if the latency is low but not if the RTT is close to 100 to 200 msec.
Non-live Streaming Video	Normally capable of sufficient host-side buffering to retransmit lost packets and tolerate moderate increases in latency. It is mainly the available bandwidth that is the major factor.
Live Streaming Video	Inherently bursty due to video compression algorithms. Applications will suffer similar issues with packet loss, latency, and jitter.
Voice over IP	VoIP is sensitive to loss, jitter, and latency similar to video.
DNS	Does not require special treatment, could be impacted if latency is very high. Traditionally DNS is UDP-based, but new DNS protocols such as DNS over HTTPS (DoH) use TCP.
Web browsing	HTTP/1.1 uses lots of parallel sessions and uses buffers. HTTP/2 will limit the number of sessions and use larger initial congestion windows; this will lead to a reduction in the buffer requirement.
Peer to Peer software	Distributing scientific data and software packages or images such as Linux distributions. No special consideration for buffering – see the Data transfer row below.
Data Centre - Distributed Compute and Storage – MapReduce, HDFS	Such applications generate TCP Incast traffic, with resulting very short oversubscription due to the synchronised answers to requests [Roy_et_al]**Roy_et_al** A short buffer is efficient, as seen in the Facebook study. The buffers can also be tuned on the server and seems more efficient.
Data transfer	As demonstrated by [Jim_Warner], large data transfers using large pipes, over long distances with a high RTT benefit from large router buffers when a 10 Gbps source sends to a 1 Gbps destination. In this case, few lost packets dramatically affect the transfer performance if RTT is high. This is a typical use case for NRENs in international projects, but the effect of packet loss is also significant for 10Gbps to 10Gbps interfaces, where just a fractional percentage loss can have a dramatic effect, especially for high RTT paths. This is also why Google developed TCP-BBR, so that TCP loss does not dramatically effect throughput in the way it does for classic TCP.

Table 2-1: Application impacted by packet loss and delay variation

The buffer memory could be inside the NPU / Forwarding ASIC or in an external memory. The former

saves space and power consumption but does not allow for very large buffers. In the latter, additional memory needs to have a large bandwidth and therefore the technical solution is expensive.

In conclusion, the data centre and backbone scenarios differ a lot. As the RTT is very low in a data centre, the buffer usage depends on the applications instantiated in there. In the DC case, buffer usage appears often even with an almost empty network at 1% or 10% utilisation, but a small buffer is enough to manage this. In telecom backbones, packet losses occurred only when utilisation was above 90% for a 30-second average. A large buffer of five msec. seemed enough and significantly efficient.

Adding delay and delay variation (jitter) impacts some applications such as VoIP or Live Streaming Video. On NREN backbones, where a long distance data transfer is happening from a high speed transfer source sending to a slower speed transfer destination, large buffers are required. While transfers may also happen between equally matched interfaces, this type of long distance large scale data transfer is a use case that is widely served by NRENs.

Large buffers have to be considered in case of QoS or oversubscribed links. Today, white boxes are available with small or large buffers (Jericho 4 GByte per ASIC), and buffer size is one of the architectural parameters that the network architects must optimise.

2.1.3 Performance Tests

To address the NRENs' concerns regarding congestion and large buffers, PSNC built a testbed to be able to demonstrate the buffering capabilities of a single white box platform. This test was led by PSNC in the context of the LSR/P router use case.

The main goal of the test was to verify whether the 'head of line' blocking and back pressure (according to [RFC2889]) appears on the tested white box platform.

For the test four 100GE interfaces were used. The white box platform was configured as an MPLS LSR in order to switch MPLS packets. On the Spirent TestCenter intermediate MPLS routers were emulated. On top of this setup, the RFC2889 Congestion Control script was started on a traffic injector (Spirent).

The test indicated that for a range of frame sizes starting from 64B to 1518B, load levels from 60 to 100% showed no head of line blocking or back pressure effects on the tested platform.

The main goal of the test was to evaluate the burst handling capabilities of the MPLS LSR router built with a white box platform and independent NOS. In the given case the Edgecore and IPinfusion devices were tested. The testbed shown in Figure 2.4 emulated the MPLS network with intermediate LSRs on the Spirent STC tester. From two 100GE interfaces traffic was sent to a single egress interface to emulate congestion conditions.

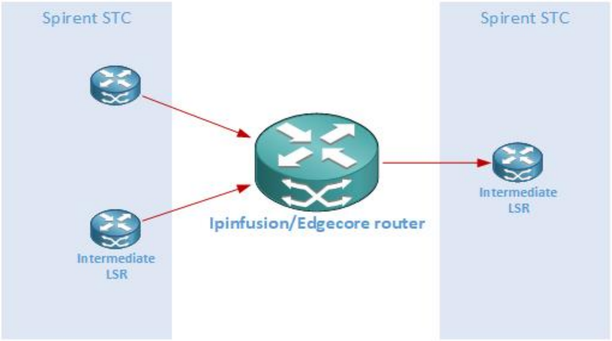


Figure 2.4: Traffic burst P/LSR testbed

The traffic was sent for 10 seconds and its characteristic was changed in incremental steps in order to measure its burst-handling performance. The source interface load was changed from 25% to 55% in steps of 5%. For each load value the number of burst packets was changed from 50,000 to 1,000,000 with a step of 50,000 packets per second (pps).

Although the actual Internet traffic mix has changed over time, the standardised IMIX profiles used for testing have not been updated accordingly because the IMIX test results need to be comparable. The IMIX packet size distribution is shown in Table 2-2.

iMIX Distribution	Frame Length Mode	IP Total Length	Default Ethernet	POS Length	Weight	Percentage (%)
Default	FIXED	40	64	64	7	58,33
Default	FIXED	576	594	594	4	33,33
Default	FIXED	1500	1518	1518	1	8,33

Table 2-2: IMIX packet distribution

The IMIX traffic was sent from two 100GE interfaces for 10 seconds to a single 100GE interface in order to generate a temporary congestion state. The tested platform was able to handle bursty traffic up to 350k PPS without packet loss when the average load on the single source interface did not exceed 45% link utilisation. At the same time, for properly switched packets, the average delay was lower than 20 µs, as shown in Figure 2.5. For larger burst sizes, the tested platform was able to handle the traffic with packet loss lower than 1%, keeping delay below 35 µs. The test results show that the platform offers line-rate switching for time sensitive applications which do not require large buffers.

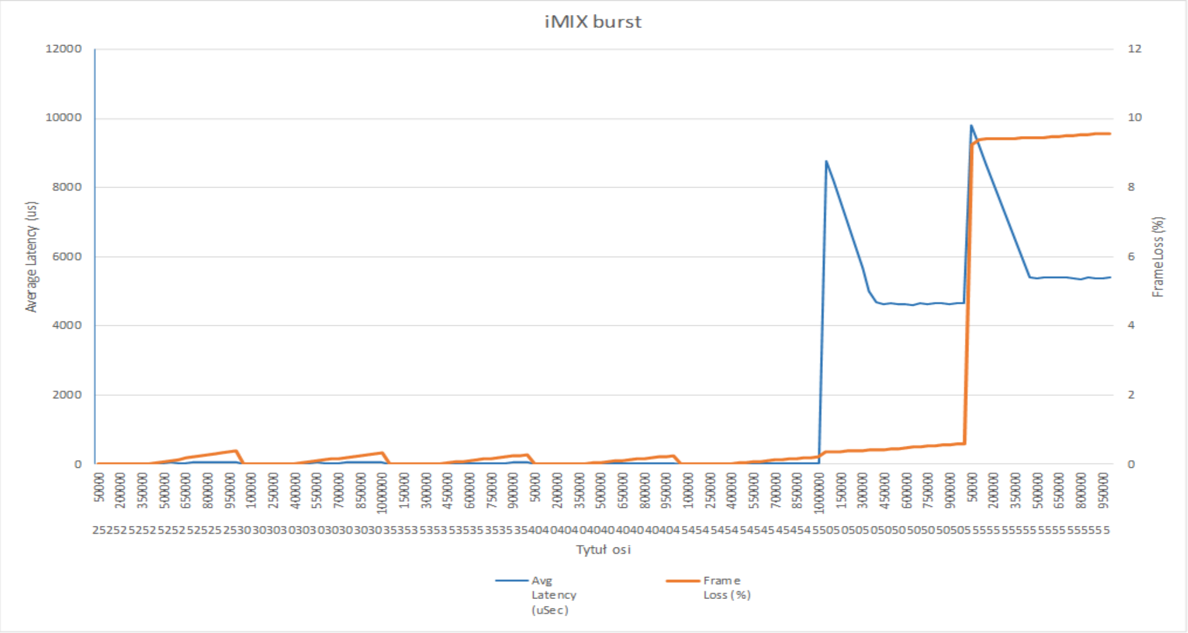


Figure 2.5: Burst impact test results

2.2 Use Cases - Selection and Work Methodology

At the beginning of the project, the WP6 T1 team selected a set of use cases that the participating NREN partners would consider realistic to implement in production. Initially the GÉANT team considered using white boxes for its LHC traffic (corresponding to a data-intensive science / big science project use case) and this use case seemed to be a promising candidate, but then the usual GÉANT network supplier proposed a new traditional solution at a very competitive price, which led GÉANT to abandon exploring this white box avenue. A vendor cutting their prices in response to the threat to their business posed by white box solutions will be a challenge for white box deployment, however, cheaper traditional solutions are also good for NRENs.

WP6 T1 is working on the following use cases:

- Customer-premises equipment (CPE)
- Provider Router (P) / Label Switch Router (LSR)
- Data centre (or cloud) fabric
- Internet eXchange point (IX)

Each of these use cases follow the below assessment process before going into production:

1. Use case specification.
2. Technical validation – switch and routing features, management features (monitoring, etc.), security features (ACL, etc.).
3. Business model (License model and TCO).
4. Qualification for production by NREN management – considering the previous analysis, NREN management will take a business decision based also on the general context (manpower availability, strategic plan, etc.).
5. Production – deployment plan.

The following section presents each of the use cases in more details, including the current work status.

2.2.1 CPE Normandy

In the region of Normandy, approximately 140 high schools are currently connected through a network using old versions of CPE routers, whose capacity is limited. The CPEs have therefore become a bottleneck, especially in cases where dark fibre is now available and needs to be renewed. The CPE specification requires the bandwidth to be increased to 1Gbps or more. Further, a list of required routing (BGP peering, IGP, VLAN, Logical interface, VRF light), management (SSH, Syslog, SNMPv2) and security (line-rate IPv4/IPv6 L3 ACLs, Broadcast storm protection) features was specified, including automation. The cost cannot exceed the cost of the existing solution. Taking into consideration that white boxes were originally designed for data centre use, even if they are cheaper than traditional CPEs, they are still not cost effective in comparison to a very small router. In this use case, the connection throughput requirement is not very high, therefore a solution based on x86 servers with a switch-style form factor is suitable. Moreover, it is possible to add additional network functions like a firewall in the future.

Figure 2.6 shows the basic architecture of the Normandy CPE. The chosen NOSs use Linux as platform, which makes it feasible to implement automation solutions based on software. This solution is very flexible, at a low cost.

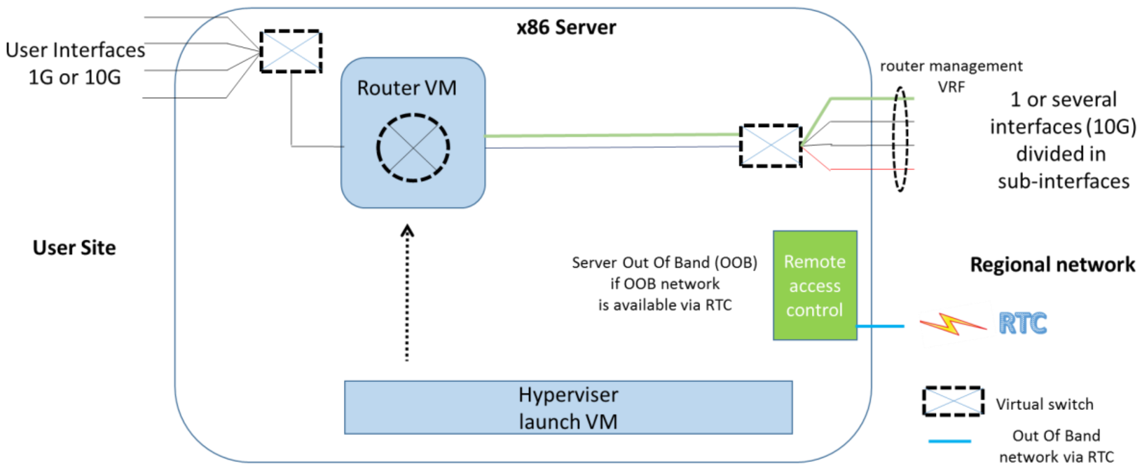


Figure 2.6: Normandy CPE architecture

Two different NOSs, Free Range Routing (FRR) and Cumulus Networks, were evaluated on a testbed. For the proposed scenario, both worked perfectly. Cumulus Networks even uses FRR as part of their NOS to handle the IP Routing and Forwarding functions. It was only in the management plane where Cumulus had advantages over FRR, providing user accounts with protocols like TACACS+ and RADIUS. From a security perspective, Cumulus supported an ACL implementation with Linux IP lists, whereas for FRR, IP tables had to be used. However, since the white box was capable of running NFVs, if stronger security features were required, an NFV firewall could be configured.

For the automation part, both NOSs integrated successfully with Ansible. In fact, Cumulus offered a module inside Ansible to make provisioning easier, facilitating command manipulation in Ansible Scripts. But even without a dedicated Ansible module, FRR could use the Linux shell module to be provisioned in an efficient way. Basically, the difference was that shell commands needed to be more specific for any given command. However, as stated, it worked successfully when integrated with FRR.

Based on the results of these tests (system and equipment management, security, routing protocols, automation deployment) and after considering the advantages and disadvantages of both solutions, the Normandy's regional network chose FRR as the solution most suitable for the project deployment in production.

For the coming production phase, the idea is to start over with two high schools connected through their own white box running an FRR NOS. Normandy's regional network has already bought the two Dell x86 servers and plans to start the first deployment (real scenario pilot test) in October 2019. Final lessons will be learned when the solution goes into production.

2.2.2 FUNET CPE (F-CPE)

In light of new nation-wide network upgrade changes, FUNET is currently working on replacing existing CPE devices, and white boxes could be an option. The motivation for this is the price of a white box and the functionalities that can be customised.

The typical dual router setup is illustrated in Figure 2.7. The FUNET White Box CPE (F-CPE) would provide uplink connectivity with BGP and routed access for different subnets in the existing L2 networks. The routed access is typically protected against single-point-of-failures using the Virtual Router Redundancy Protocol (VRRP). There might also be a customer-owned firewall which is usually installed between the edge routers and L2 networks. If there are any special user requirements like the need to bypass the firewall, those users can be connected directly to the F-CPE routers.

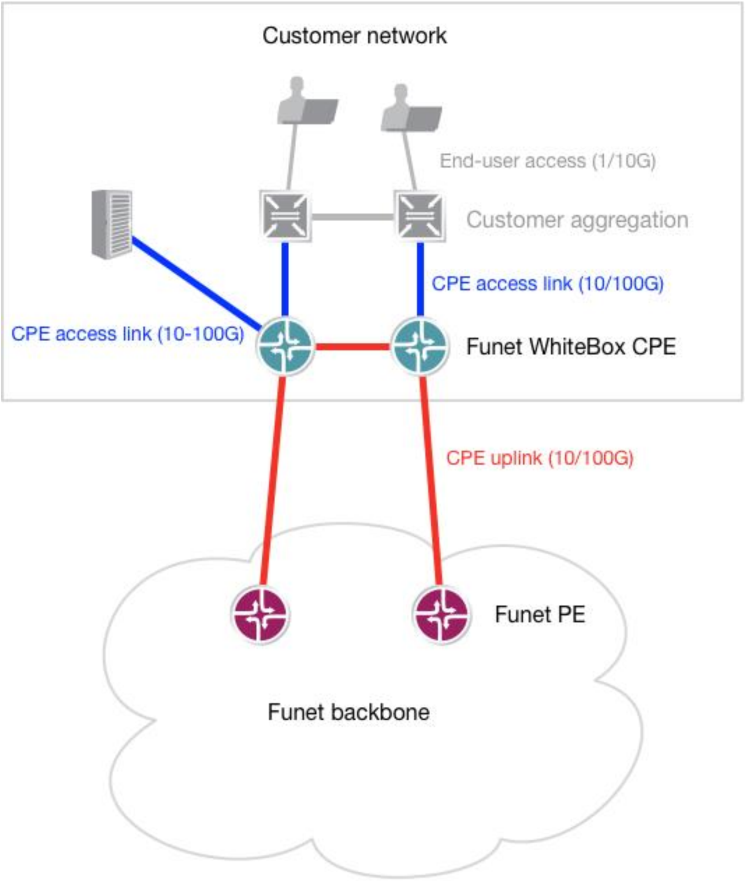


Figure 2.7: FUNET CPE project

The F-CPE use case was tested in a VMware environment with a Cumulus VX NOS appliance to evaluate control plane support for the required features. Cumulus provided commercial software support.

Feature tests were performed with two NOS instances connected directly to the FUNET backbone with BGP. Client connectivity was evaluated with a separate Ubuntu Linux virtual machine. The existing infrastructure was used for management, syslog, monitoring and DHCP/DHCPv6 tests.

The following control plane features were successfully implemented:

- eBGP peering towards the FUNET backbone (both IPv4 and IPv6 unicast).
- iBGP peering between the Cumulus NOS instances (both IPv4 and IPv6 unicast).
- BGP route filtering.
- OSPFv2 (IPv4 unicast) and OSPFv3 (IPv6 unicast) as an IGP.
- VRRPv3 (IPv4 and IPv6) towards the client network.

- Loopback and interface access control lists (IPv4 and IPv6).
- Management VRF support.
- IPv4 DHCP relay support with a client host.
- IPv6 stateless address auto configuration (SLAAC).
- IPv6 DHCPv6 relay support with a client host.
- Jumbo frame support (IP MTU up to 9000 bytes).

Various management and monitoring features were also evaluated successfully:

- SSH management access with key-based authentication.
- Remote syslog (via a management VRF).
- DNS and NTP (via a management VRF).
- SNMPv2 polling and feeding interface counters to InfluxDB/Grafana (via a management VRF).
- Configuration backup and restore.
- NOS software upgrade.

The control plane and BGP dynamic routing were observed to be stable for months, and the control plane was responsive. The tests performed show that the main control plane functionality needed in the CPE use case is supported by Cumulus, so that the NOS can be considered for that purpose. In the next phase the Cumulus NOS should be tested in a real hardware environment to evaluate its forwarding plane performance, and control plane and forwarding plane interoperability. All these tests and a cost analysis should allow FUNET to decide whether to proceed towards production or not. In the CPE use case implemented with a hardware white box, the biggest challenge would be finding hardware which provides the port density needed in a CPE environment while being cost-effective.

2.2.3 GRNET Data Centre

In this use case, a white box solution is evaluated in a cloud fabric context with the expectation to gain the benefit of cost-reduction, following a large-scale cloud provider example like Facebook, with hardware-NOS decoupling, less vendor lock-in, avoiding proprietary solutions, shorter life cycles, with fully automated management and service provisioning.

For the DC use case, speeds beyond 10 Gbps for the server ports are desirable, keeping a speed of 10 Gbps as the minimum requirement for server-facing interfaces.

The most typical setup for the server ports is a dual ToR switch with link connectivity that enhances reliability requirements, provides more flexibility on the support side (e.g., ToR switch upgrades without traffic interruption) and doubles the link capacity. In this setup each server is connected with 1x10 Gbps to the same rack ToR switch and also using cross-rack cabling to the adjacent rack ToR switch with another 1x10 Gbps. Basic hardware redundancy options like dual power feeds and hot-swappable power supplies and fan units might also be beneficial. The basic node redundancy includes a non-redundant control plane, hot-swappable power supplies, hot-swappable fan units and next business day support service.

The DC fabric use case closely follows GRNET's current DC architecture, as shown in Figure 2.8. The basic service is VLAN provisioning to customer/server ports on ToR switches and customer/server multi-homing on two adjacent ToRs using EVPN/VXLAN Ethernet segment mechanisms and LACP for bonding. The mandatory features tested are EVPN/VXLAN Ethernet segment mechanisms and LACP, Ethernet interface setup, BGP/EVPN and VXLAN, SSH, DNS, NTP, Ansible, NETCONF (but not VRFs in the first phase) and multicast. Another issue under investigation is the potential use of spine switches as DC routers, running BGP protocol for inter-DC and IP network interconnections, as shown in Figure 2.8. Therefore, ACLs are going to be tested for the L3 interfaces, evaluating their overall size and the number that can be supported by white box devices. The entire configuration should be manipulated by an automation mechanism such as Ansible for configuration creation and NETCONF for configuration deployment.

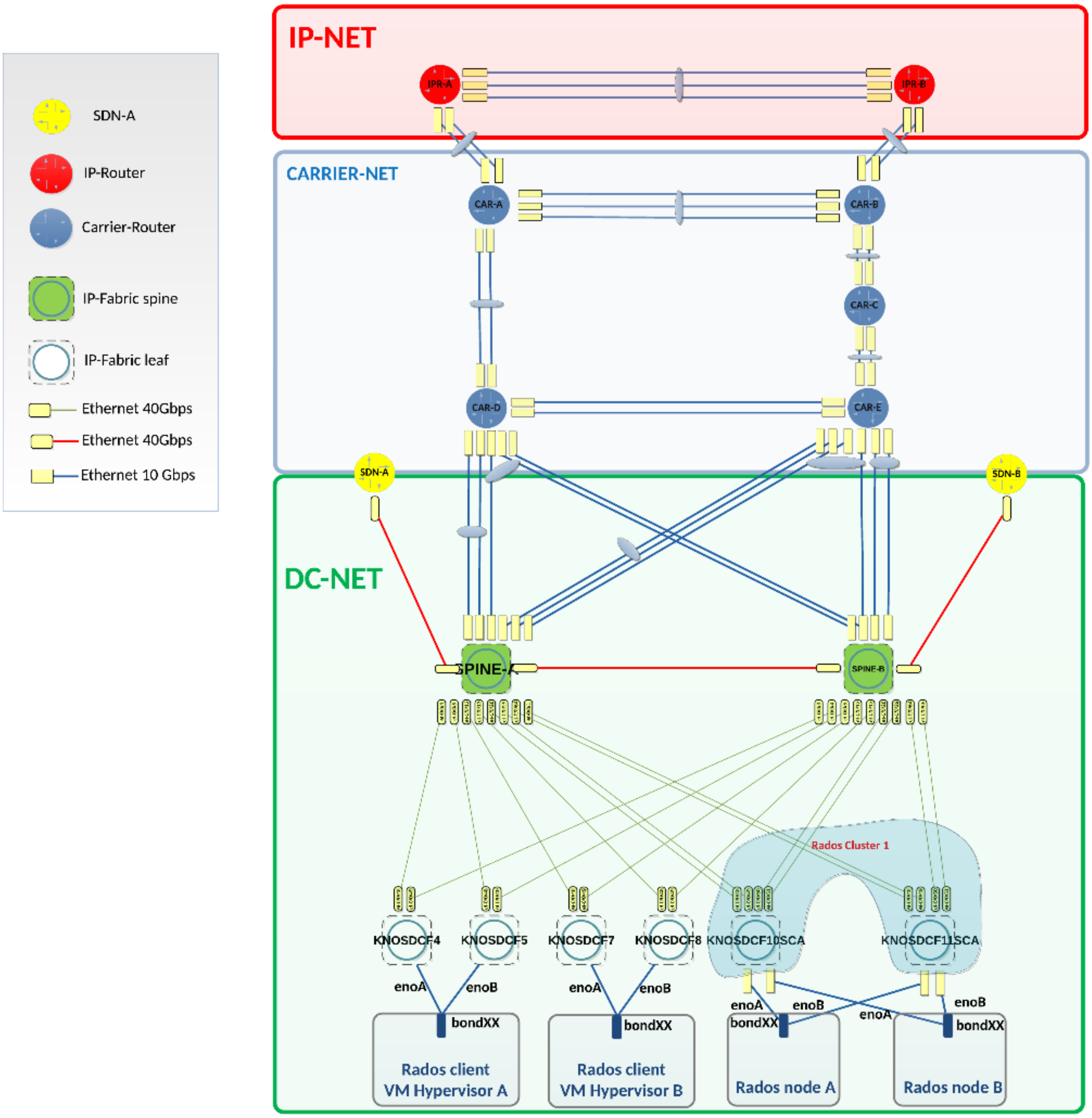


Figure 2.8: GRNET data centre project

Currently the fully-fledged white box-based DC fabric network laboratory is under a procurement process. GRNET used a virtual environment, based on the EVE-NG platform, to test an EVPN/VXLAN-based control plane on top of the expected Clos topology. Virtual spine and leaf switches running Cumulus OS were deployed and the result is a fully functional environment. The management plane was successfully handled thanks to NETCONF. The next round of tests (undergoing) is related to the EVPN control-plane and the routing models, including asymmetric, symmetric or centralised routing. An evaluation of the troubleshooting methods/tools is also planned for this stage.

2.2.4 Normandy Data Centre

The Normandy regional network wants to renew and increase the size of its data centre (from 50 racks to 150 racks). It was decided to implement a first slice of ten racks as a prototype to assess the solution in a real operational environment.

It was expected that white boxes would have the benefits of avoiding vendor lock-in and reducing CapEx. However, a tender was published and the result showed that traditional vendors answered with very competitive proposals. The interesting point here is the rationale leading to the regional network’s decision. Taking into consideration that the traditional vendor’s offer was at the same level as the white box vendors' offer, and the fact that the white box would require new skills sets that would need to be learnt, the white box total cost of ownership (TCO) was judged to be higher than the traditional vendor’s TCO. Therefore, it was decided to choose the traditional vendor solution. This example suggests that traditional vendors have some margin to reduce prices to try to retain customers and market share; the interesting question is whether that is sustainable, especially as the white box market matures.

2.2.5 Provider Router (P) / Label Switch Router

The objective in this use case is to deploy efficient and cost effective LSR routers in the GÉANT or NREN MPLS backbone. An LSR router allows PE router traffic aggregation and transports PE traffic between PoPs. LSR routers in most cases do not terminate particular services. There is currently a strong need for 100G and 400G interfaces in core networks. The scalability of MPLS networks can be increased by adding another level in the network hierarchy in the middle of the network where additional core devices can be installed, and be used to aggregate and exchange the traffic from PE devices, as shown in Figure 2.9 and Figure 2.10. These new devices constructing a ‘super-core’ of the network must provide very high switching capacity but do not need to implement the wide set of features needed by PE devices. Moreover, the ability to install optical modules supporting selectable wavelength transmission increases the potential capacity of fibre links. The number of required transponders increases as the capacity of the network grows. Each time a pair of transponders is required, it increases CapEx and OpEx. In addition, LSR devices are more efficient when it comes to power consumption, thus reducing OpEx.

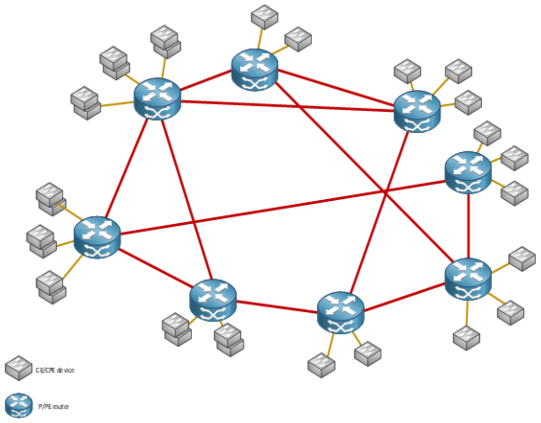


Figure 2.9: Collapsed core architecture

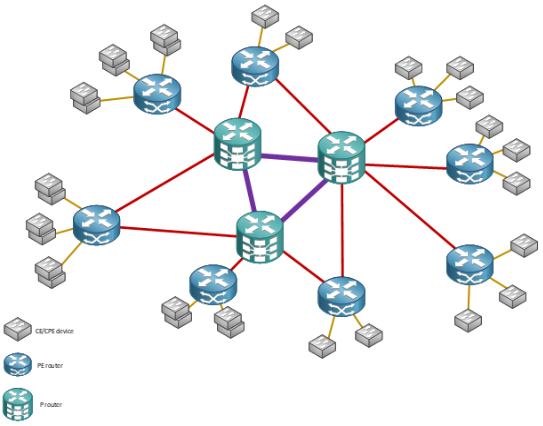


Figure 2.10: P/LSR core architecture

This concept was set up in laboratory environment. The main purpose of the test was to verify the basic LSR functionalities of the white box platform. The Spirent TestCenter application and an N11U chassis with 100G and 100G interfaces were used. The IPinfusion NOS was implemented on a white box device (EdgeCore EC_AS5912-54X) and configured as an MPLS LSR. On the Spirent TestCentrer application, the PE and intermediate LSR routers were emulated. The LSPs were established on the testbed (see Figure 2.11) between pairs of PE routers, passing through the device under test.

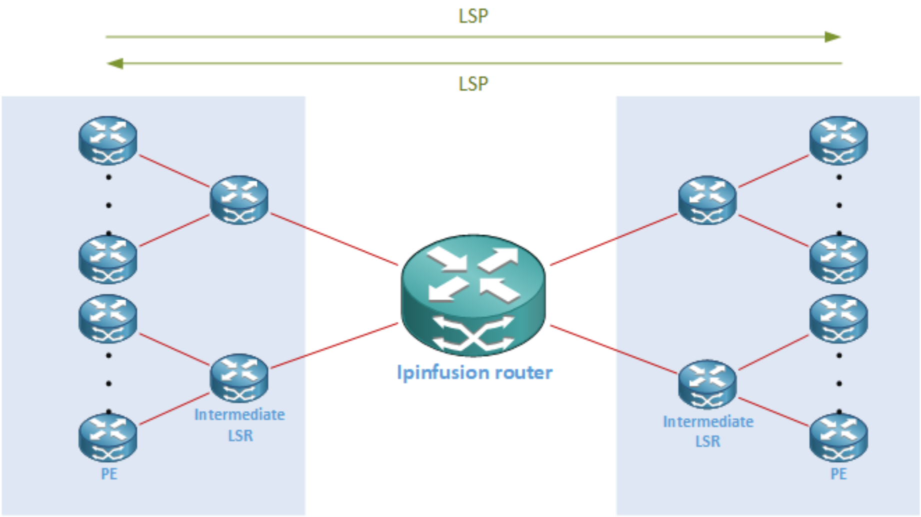
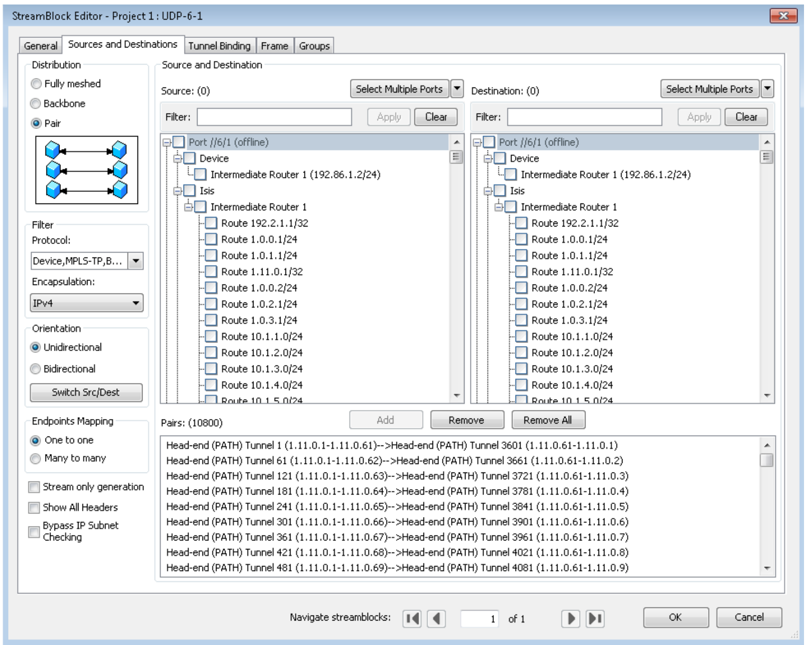


Figure 2.11: LSR/P testbed

Once configured on the tester the 10800 LSP paths were signalled. Next the traffic stream was generated to verify the MPLS forwarding plane operation. During the test, all LSPs were up and no traffic loss was observed, as shown in Figure 2.12.


StreamBlock Editor - Project 1 : UDP-6-1

General | Sources and Destinations | Tunnel Binding | Frame | Groups

Distribution: ○ Fully meshed ○ Backbone ● Pair

Filter
Protocol: Device,MPLS-TP,B...
Encapsulation: IPv4

Orientation: ● Unidirectional ○ Bidirectional

Endpoints Mapping: ● One to one ○ Many to many

□ Stream only generation
□ Show All Headers
□ Bypass IP Subnet Checking

Source and Destination

Source: (0)

Select Multiple Ports
Filter:
Apply
Clear

```
□ Port //6/1 (offline)
  □ Device
    □ Intermediate Router 1 (192.86.1.2/24)
  □ Isis
    □ Intermediate Router 1
      □ Route 192.2.1.1/32
      □ Route 1.0.0.1/24
      □ Route 1.0.1.1/24
      □ Route 1.11.0.1/32
      □ Route 1.0.0.2/24
      □ Route 1.0.2.1/24
      □ Route 1.0.3.1/24
      □ Route 10.1.1.0/24
      □ Route 10.1.2.0/24
      □ Route 10.1.3.0/24
      □ Route 10.1.4.0/24
      □ Route 10.1.5.0/24
```

Destination: (0)

```
□ Port //6/1 (offline)
  □ Device
    □ Intermediate Router 1 (192.86.1.2/24)
  □ Isis
    □ Intermediate Router 1
      □ Route 192.2.1.1/32
      □ Route 1.0.0.1/24
      □ Route 1.0.1.1/24
      □ Route 1.11.0.1/32
      □ Route 1.0.0.2/24
      □ Route 1.0.2.1/24
      □ Route 1.0.3.1/24
      □ Route 10.1.1.0/24
      □ Route 10.1.2.0/24
      □ Route 10.1.3.0/24
      □ Route 10.1.4.0/24
      □ Route 10.1.5.0/24
```

Pairs: (10800)

```
Head-end (PATH) Tunnel 1 (1.11.0.1-1.11.0.61)-->Head-end (PATH) Tunnel 3601 (1.11.0.61-1.11.0.1)
Head-end (PATH) Tunnel 61 (1.11.0.1-1.11.0.62)-->Head-end (PATH) Tunnel 3661 (1.11.0.61-1.11.0.2)
Head-end (PATH) Tunnel 121 (1.11.0.1-1.11.0.63)-->Head-end (PATH) Tunnel 3721 (1.11.0.61-1.11.0.3)
Head-end (PATH) Tunnel 181 (1.11.0.1-1.11.0.64)-->Head-end (PATH) Tunnel 3781 (1.11.0.61-1.11.0.4)
Head-end (PATH) Tunnel 241 (1.11.0.1-1.11.0.65)-->Head-end (PATH) Tunnel 3841 (1.11.0.61-1.11.0.5)
Head-end (PATH) Tunnel 301 (1.11.0.1-1.11.0.66)-->Head-end (PATH) Tunnel 3901 (1.11.0.61-1.11.0.6)
Head-end (PATH) Tunnel 361 (1.11.0.1-1.11.0.67)-->Head-end (PATH) Tunnel 3961 (1.11.0.61-1.11.0.7)
Head-end (PATH) Tunnel 421 (1.11.0.1-1.11.0.68)-->Head-end (PATH) Tunnel 4021 (1.11.0.61-1.11.0.8)
Head-end (PATH) Tunnel 481 (1.11.0.1-1.11.0.69)-->Head-end (PATH) Tunnel 4081 (1.11.0.61-1.11.0.9)
```

Navigate streamblocks: 1 of 1

Figure 2.12: LSR/P testbed

The same setup was also used for the buffer size and performance validation (see Section 2.1.2.).

At this stage, the investigation shows that the OcNOS operating system offers a basic set of MPLS features allowing an MPLS network to be built and services provided on it. It has to be noted the software is still in an early development phase, so the network administrator has to check the supported features and the software development roadmap in order to make a final decision. It is very important to perform interoperability tests in cases where the white box platform has to be connected to an existing network. For the moment some protocol or hardware related options have default values which cannot be changed by an administrator. Therefore, some final parameter adjusting must be tested before production deployment.

2.2.6 Internet eXchange Point (IX)

The main purpose of a Global Internet eXchange point (GIX) is to connect multiple entities in a dedicated architecture and enable the creation of direct peerings between them. The most common architecture is layer 2 with switches that provide connectivity between all the customer routers in the same network, or with a route server service that eases the establishment of customer BGP peerings.

This white box use case has been selected to test if this new concept can fulfil all the requirements of RENATER's Internet Exchange point (SFINX) and at the same time provide a cheaper solution. OcNOS by IPInfusion was chosen as the NOS, as it is used in production by LYNX, a London GIX.

The testbed included two white boxes and one Juniper MX104 with a logical system to simulate a route reflector and the clients that send routes and perform BGP peering. Open Network Install Environment (ONIE) and all the required features were tested and Table 2-3 summarises the essential features.

Plane	Example of tested features
ONIE	Remote and automatic installation (DHCP, Web server), manual installation.
Management plane	SNMP, TACACS, RADIUS, LOGS, NETFLOW, SSH, etc.
Control plane	OSPF, VLAN, RSTP VXLAN, etc.
Security	MAC/IP ACL, BPDU filters, etc.
Data plane	MAC address table limitation, ARP table limitation.

Table 2-3: Excerpt of GIX features testbed

All tests were completed successfully, the results were conclusive and validated all the required features. Some non-blocking limitations were also identified, such as the maximum number of characters on an interface description, which is limited to 32 characters. The NOS and hardware costs were significantly lower than the cost of the current set up. Therefore, it was decided that the white box solution will replace the existing solution in production. The transition will require detailed preparation to ensure minimal downtime. The transition, which will be performed by RENATER, is planned to be completed before the end of the year.

3 Data Plane Programmability

There has been a continuous effort to innovate in networking, starting from the control plane. SDN has allowed the behaviour of the control plane in switches to be programmed. Separating control from switching and centralising the control function has allowed new behaviours to be defined based on the content of standard packet headers. Modifications are implemented in the controller and sent to selected switching nodes in the form of switching rules.

However, the network chips operating in the switch's data plane are still proprietary and mostly bound to fixed switching operations implemented in silicon during the design phase.

A programmable data plane transforms the network ASICs, allowing the programming of new forwarding behaviours in the packet processor itself. Full programming control on processor memory and functions permits almost complete freedom on packet header processing, including information insertion, change and removal. Data plane programming requires adequate hardware, like FPGA or Barefoot Tofino [Barefoot], however the cost is not significantly higher than traditional Ethernet switching hardware.

The advantages are (efficient) silicon programming using languages like P4, extreme flexibility in packet handling (telemetry), new function/protocol inclusion without hardware replacement, while maintaining wire speed performance. The effort on the Tofino chip is being standardised in the Protocol Independent Switch Architecture (PISA) by Barefoot (now Intel).

3.1 Use Cases

Two use cases are selected for the work on DPP: In-Band Network Telemetry (INT) and Distributed Denial of Service (DDoS) detection and mitigation.

3.1.1 In-Band Network Telemetry

Extending the set of supported actions, protocols and monitoring approaches requires software tools. Unfortunately, the software performance is not sufficient as most devices today are limited to only sampling the observed traffic for high-speed flows. This is a drawback especially in a high-speed network environment like NREN backbones. Therefore, the capability of programmable monitoring in a P4 white box is very interesting because it allows easy deployment of new monitoring approaches at the speed of the network line. Data plane programming allows also the insertion of additional information in each packet header, to be removed transparently later by an enabled node, and the sending of data to an external elaboration engine.

WP6 T1 is testing the feasibility of this approach with In-Band Network Telemetry (INT) [INT] which is an emerging standard for real-time network monitoring. INT technology is based on the insertion of telemetric data into each passing frame as presented in Figure 3.1, with departure and arrival time fields. The telemetric data can carry information like the occupancy of switch queues, the time required for the crossing of autonomous systems or security-related information. For example, it is possible to indicate malicious traffic with invalid protocols/protocol field values or traffic which is suspected to be DDoS traffic. This information can be added and processed directly at the data-plane layer and it can be used later in software which can perform deep packet inspection.

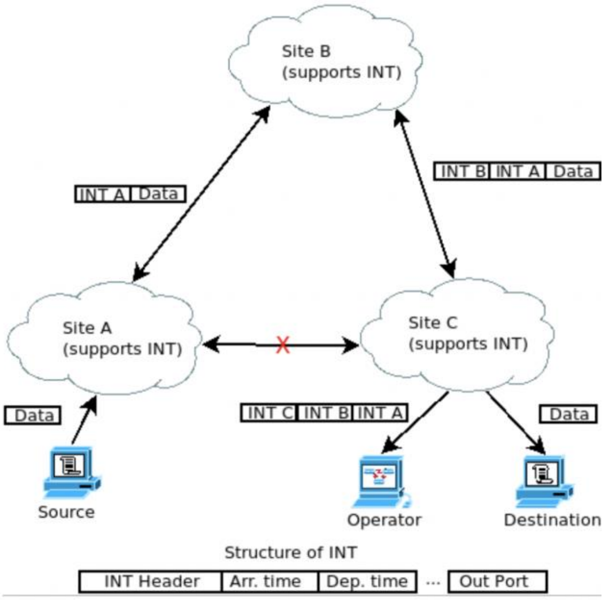


Figure 3.1: INT testbed plan

WP6 T1's work aims to provide a precise measurement of network flows, without sampling, based on the initial parameters: delay, delay variation, packet loss and reordering. It will be performed with a standard x86-64-based server with PCIe and Linux using FPGA cards (two 100G ports and powerful Virtex UltraScale+ FPGA [Liberouter]). The Linux drivers support DPDK [DPDK] and 200Gbps transfers into RAM memory. The measurements can be used by a low latency application (such as the LoLa project being considered elsewhere in WP6 T1) in a multi-domain environment. The telemetry use case is based on CESNET work related to P4 INT which was previously presented during the P4 workshop at Stanford, CA, USA [CESNET]. CESNET also developed a compiler of the $P4_{14}$ language to VHDL which is one of the HDL languages used for the description of digital circuits. This allows the implementation of a processing pipeline described in a P4 program. The generated hardware is capable of processing network data at a speed of 100 Gbps. More details about the generated architecture are available online [Benycze].

So far, the P4 INT design defined by WP6 T1 was ported to the newest FPGA-based SmartNIC developed at CESNET. The design is capable of managing a throughput of 100 Gbps. The P4 INT pipeline is used on both network interfaces, so the network card is capable of processing network traffic at 200 Gbps.

WP6 T1 has started to work on the new compiler implementation which generates VHDL code from a provided $P4_{16}$ program, while the older $P4_{14}$ version will also be supported.

3.1.2 DDoS Detection

DDoS is a constant threat to NREN users and network services. This use case has the objective to validate data plane programming with P4 as a tool for service improvement in terms of responsiveness and precision. In the first phase, the goals of the DDoS detection activity are very fast detection of DDoS attacks on boundaries of NRENs/GÉANT networks and provision of detailed information about DDoS traffic characteristics as shown in Figure 3.2. In phase two, the goal is to achieve almost immediate mitigation of the attack. In the activity, the WP6 T1 team has been implementing DDoS traffic detection and monitoring directly in a P4 programmable switch with use of big data streaming sketch memory structures, as described below.

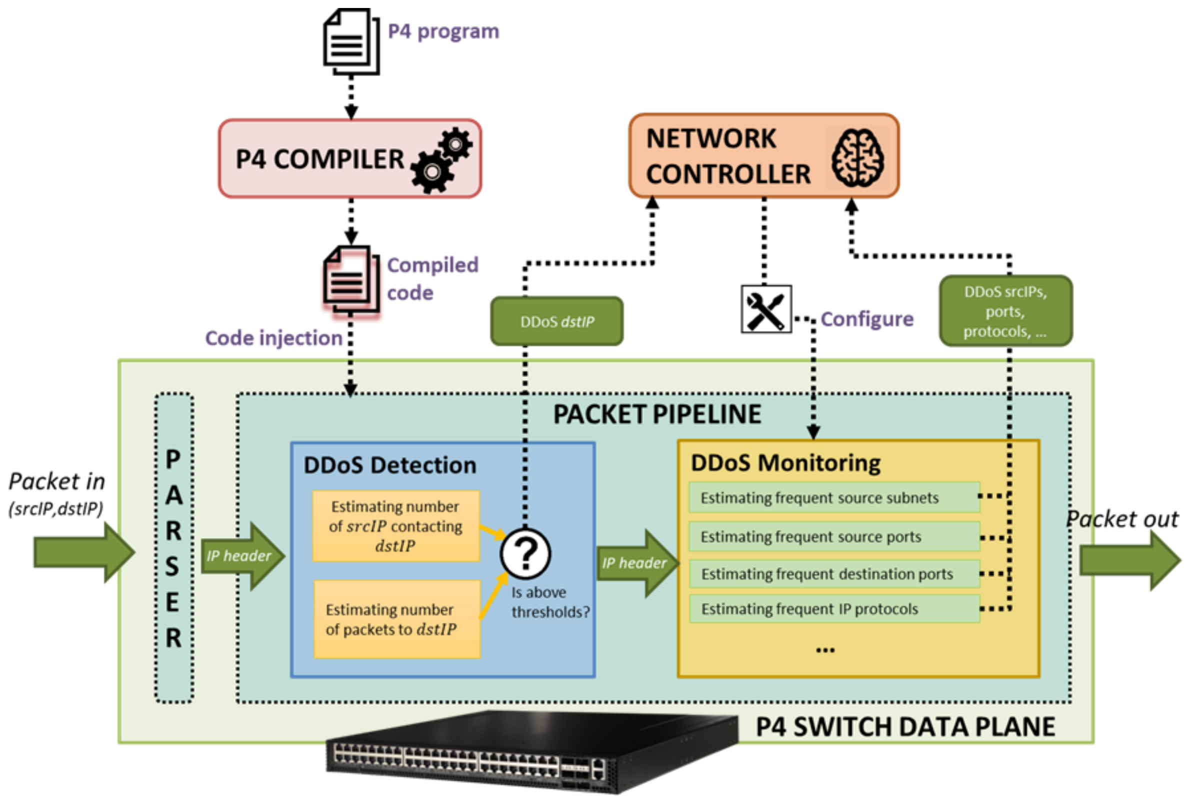

P4 program
P4 COMPILER
NETWORK CONTROLLER
Compiled code
Code injection
DDoS dstIP
Configure
DDoS srcIPs, ports, protocols, ...
PACKET PIPELINE
Packet in (srcIP,dstIP)
PARSER
IP header
DDoS Detection
Estimating number of srcIP contacting dstIP
Estimating number of packets to dstIP
?
Is above thresholds?
IP header
DDoS Monitoring
Estimating frequent source subnets
Estimating frequent source ports
Estimating frequent destination ports
Estimating frequent IP protocols
...
Packet out
P4 SWITCH DATA PLANE

Figure 3.2: Overview of the DDoS detection and monitoring prototype

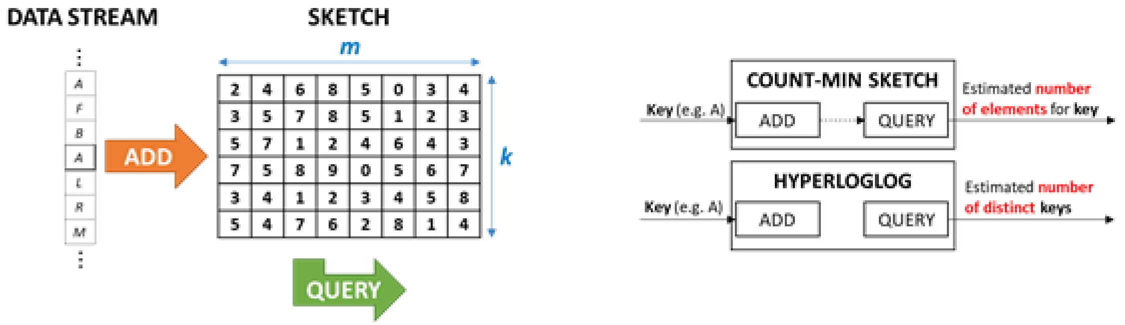
The DDoS detection and monitoring prototype provides important information about active incoming DDoS attacks and can greatly enhance countermeasures against this type of attack. The key benefit of using P4 switches for handling DDoS traffic is that detection algorithms can be adapted for new types of DDoS attacks and added to the switch in quite a short amount of time.

3.1.2.1 *Sketch Structures for DDoS Traffic Detection and Monitoring*

DDoS detection and DDoS traffic monitoring can be performed directly at the data plane level of the white box thanks to the use of big data streaming sketch memory structures. The sketch structures provide memory-efficient collection of summarised traffic statistics and have some interesting benefits in comparison to the currently used monitoring techniques. They process packets at full wire speed, performing a set of actions for every packet. Moreover, all processed packets can contribute to traffic statistics without any performance penalty. This is not true for the most popular current approaches, sFlow and NetFlow, which require packet sampling (for example, for 10 Gbps, sampling might typically be at a rate of about 1 packet for every 2000 packets [SFLOW_sampling]) and which only consider the packets they review. This is not ideal for fast DDoS detection.

Another benefit is that the sketch structures provide only aggregate information and require low volume communication between the network node and the network controller. This can be compared to Netflow, which is another monitoring technique. NetFlow sends information about every detected 5-tuple flow, which means that in certain situations a lot of information will be passed to the network controller. This can be problematic when a lot of short-lived flows are detected as in the case of a typical DDoS attack. The sketches do not require any additional CPU-intensive analytical processing, which is mostly performed on powerful server machines in comparison to NetFlow and sFlow.

In general, sketch structures (see Figure 3.3) can provide a statistical estimation of an item's frequency in a big data stream (provided by a Count-min sketch), an item's cardinality (through a HyperLogLog algorithm) or its membership (when a Bloom filter is used). The simplicity of sketch structures, which are mostly based on hash algorithm operations, allows the implementation of sketch algorithm logic in P4 programs which can be deployed on programmable white box switches.



2	4	6	8	5	0	3	4
3	5	7	8	5	1	2	3
5	7	1	2	4	6	4	3
7	5	8	9	0	5	6	7
3	4	1	2	3	4	5	8
5	4	7	6	2	8	1	4

Figure 3.3: Sketch structure

When sketch structures are applied to network traffic monitoring they can generate aggregate information on how many source IP addresses contacted a given destination IP address, how many packets were processed for each destination IP address, how many packets for each source UDP or TCP port were transmitted, and for many other properties of observed traffic.

The current WP6 T1 implementation includes two sketch structures for DDoS detection. It detects all destination IP addresses that might be subject to a DDoS attack and reports them to the network controller. A destination IP address is considered to be under attack if it is contacted by a high number of source IP addresses (above the configured threshold value) and has received a high number of packets (another configured threshold) within a short time interval. The sketch-based DDoS detection algorithm uses a novel data structure that combines Count-min and HyperLogLog sketches (see Figure 3.4) with the aim of estimating the number of distinct source IP addresses that send at least one packet to a specific destination address. A P4 implementation of this complex, multi-dimensional sketch is a challenging task.

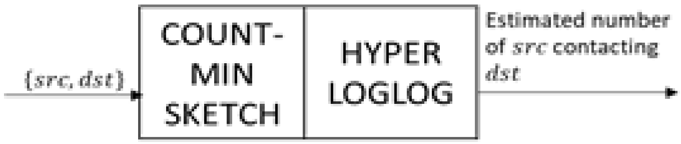


Figure 3.4: New sketch structure for the detection of DDoS attack targets

When the network controller is notified of the DDoS attack target at a specific destination IP address, the controller can activate one or more DDoS monitoring algorithms which provide more detailed DDoS traffic statistics:

- The total traffic (packets and bytes) towards the DDoS target.
- The most frequent source IP subnets (i.e. with an IPv4 /16 prefix) originating the attack.
- The most frequent source TCP/UDP ports - specific port numbers can suggest the use of DDoS amplification techniques based on vulnerable public network services (e.g., DNS, NTP, SNMP).
- The most frequent destination TCP/UDP ports which tell what service is under attack (e.g., a web portal).
- The IP protocols used (whether it is a UDP- or TCP-based DDoS attack).
- The most frequent packet lengths (amplification attacks tend to use big packets).

Figure 3.5 presents the DDoS detection workflow and DDoS monitoring algorithms deployed on the programmable data plane device.

More DDoS monitoring characteristics can be added in the future, such as the level of packet fragmentation, TCP flags used, etc.

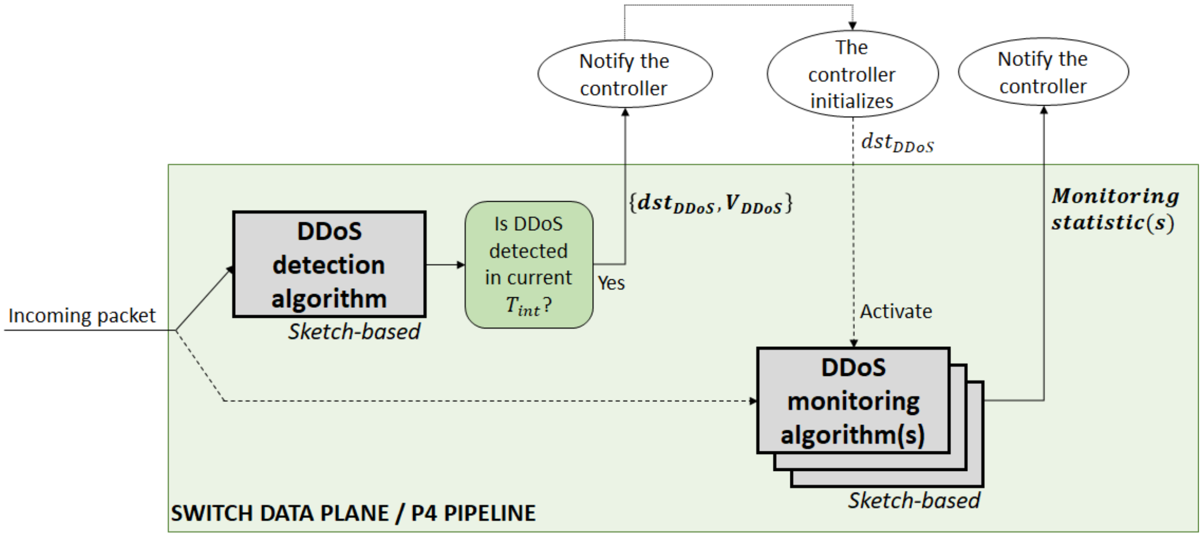


Figure 3.5: DDoS detection and DDoS monitoring workflow in a programmable data plane device

A DDoS detection and monitoring service offered by this approach could provide immediate notifications when DDoS attacks start or cease, together with frequently updated information about the characteristics of the DDoS traffic, which can give insight into the phase of the attack.

3.1.2.2 *P4 Implementation*

The initial version of DDoS detection and monitoring functionality has been implemented with P4 switches. In the first phase, a P4 behavioural model (BMv2) is used instead of white box switches. BMv2 is a P4 software switch emulator which is widely used for developing and testing P4 programs. The WP6 T1 team is using a simple virtual network topology composed of three interconnected P4 switches. Each P4 switch is connected to a host. This topology (see Figure 3.6) can be deployed on any developer machine with the use of a Docker container called p4app which contains a Mininet [Mininet] environment extended with BMv2 instead of the default Open vSwitch [Open vSwitch]. When the Docker container starts, the P4 program code is loaded into the P4 switches, sketch structures are allocated within each P4 switch and the network controller process starts. The DDoS traffic can then be generated by a Python script activated after logging to a Mininet host.

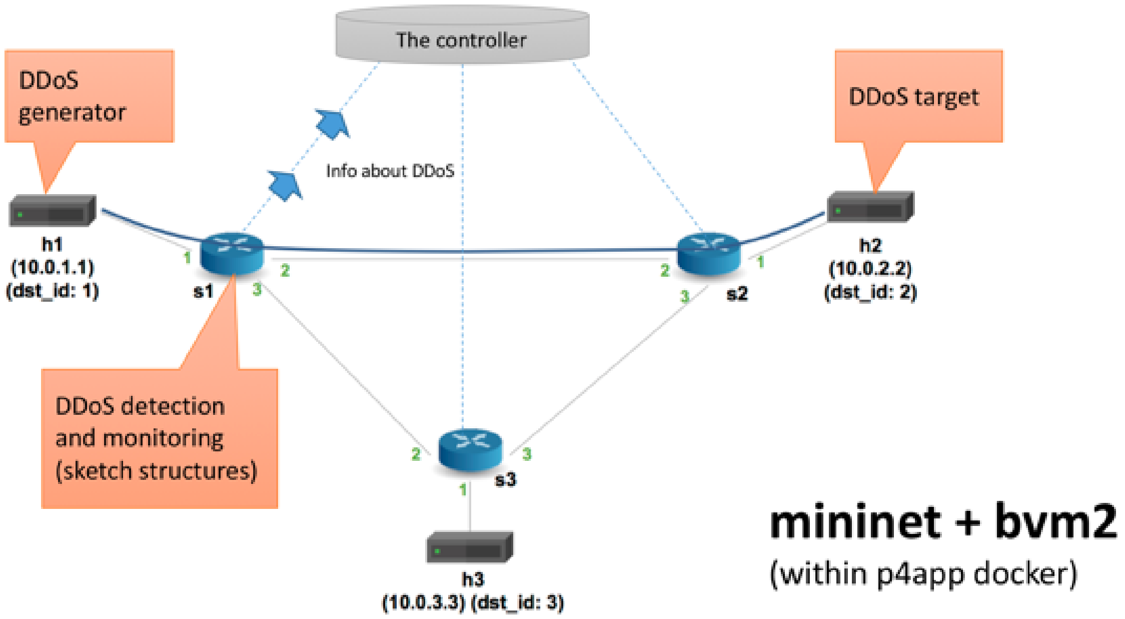


Figure 3.6: The virtual environment used for DDoS use case development

The WP6 T1 team started with sketch structures implemented in P4$_{14,}$ then upgraded the prototype to P4$_{16}$. Apache Thrift was initially used and a simple_switch_CLI, as a way to communicate with the network controller and P4 switches. In the virtual testbed, the controller was very slow: ~0.1 sec for reading a few values from a register, ~0.3 sec for adding a table entry and ~0.1 sec for resetting a single register or table. However, this basic approach allowed the team to develop and test some code. Finally, the network traffic was observed for three seconds and then five seconds were required for performing all actions between the controller and switches. The prototype development also showed that for each new instance of the sketch structure a repetition of the same P4 code is needed. If P4 could support the passing of references or pointers to P4 register structures between the control functions then it would allow developers to only need to write sketch logic code once. Another limitation of P4 is the lack of logarithm operations, which is why the implementation of the HyperLogLog-based sketch is cumbersome in P4.

The virtual testbed is now being moved to two physical testbeds, one in PSNC Poland (based on two Arista 7170-32c devices) and the second in FBK Italy (based on three Edgecore Wedge100BF-32X devices connected in a triangle). The WP6 T1 team is trying to adapt the existing P4 code for the Tofino chipset, which introduces a new set of P4 language constraints. The P4 Runtime communication performance on Tofino should be much faster; it allows the WP6 T1 team to set an observation time interval below 1 second, which in practice corresponds to how fast the DDoS detection can be triggered. Given a successful validation of this DDoS prototype, P4 switches with the prototype code could be implemented at selected locations at the borders of the PSNC and GARR production network domains. However, for a P4 switch with uploaded P4 DDoS detection code to be able to fully replace current NREN switches, more work is needed on the integration with the network protocol stack features which can be found in most of the production switches and routers (i.e., VLANs, VXLANs, MPLS, IPv4/IPv6 routing, OSPF, IS-IS, VPNs, etc.). An alternative approach however is that a P4 switch running sketches can receive mirrored traffic from one or many inter-domain links (and it would be enough to copy only the beginning of each packet in order to access IP and UDP/TCP headers).

4 Router for Academia Research and Education (RARE)

The Router for Academia, Research and Education (RARE) project combines the white box work detailed in Section 2 with the data plane programmability work of Section 3 with the goal to create a router with all the functionalities needed for the academic, research and education community. RARE aims to assess and validate different pieces of software for the control plane, the data plane, and the communication between them that will work on top of a white label hardware. The validation is done from the perspective of the use cases that might be of interest to NRENs.

The recent GN4-3 WP6 White Boxing workshop has shown that the community is interested in RARE. More than 84% of the participants answered that they would like to test the outcomes of the RARE project (see Figure 4.1).

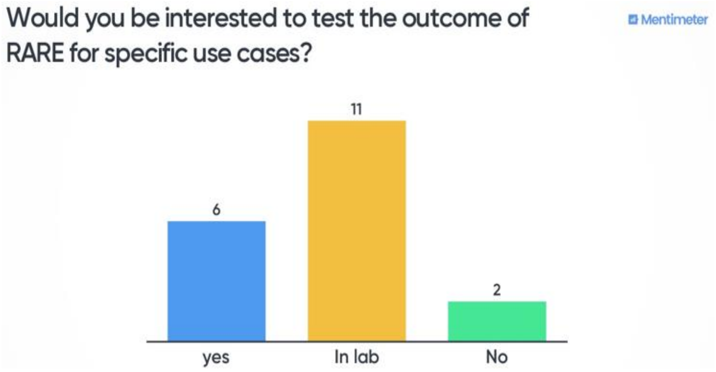


Figure 4.1: NREN survey results

A key part of the work consists of establishing control plane software to drive a data plane via a programmatic interface. P4 is a natural choice for a language to achieve this. The P4 core language tries to be as independent as possible from the target or NPU processor architecture, although some level of architecture dependence is still prominent. For now, the WP6 T1 team has chosen to use the Tofino Barefoot chipset that is available on different switches.

FreeRtr is a good first control plane candidate [FreeRtr], and has been used for years by KIFU, the Hungarian NREN. It is used as an operational route reflector but it can be used to implement an LSR router and even LER functionality. Several features will be investigated for the production stage, including monitoring and security.

The WP6 T1 team will consider the following use cases according to their implementation simplicity and their chances to be deployed to production:

- A baseline feature set common to all use cases below: SSH transport for management, TACACS/RADIUS for management, infrastructure ACLs to protect router interfaces (also known as CoPP), LPTS, CP protection, monitoring capability providing link utilisation and CPU counters if relevant.
- Service Provider grade P router (the P function relates to the capacity of a router to only switch traffic at a high line rate): IPv4/Ipv6 addressing, IS-IS (or OSPF) IGP routing, MPLS/LDP, Segment Routing over MPLS|IPv6.
- Telecom Service Provider Edge grade PE router with a minimum feature set: L3VPN (IPv4 might be sufficient as a start), L2VPN (EVPN might be sufficient as a start), point to point, point to multipoint, multipoint to multipoint.
- Performance validation: Full line rate performance, table size scalability performance.

The code is being validated using the BAREFOOT bf_switchd virtual switch. Technically everything can be developed in this virtual environment. However, high-scale data plane throughput can only be tested with real hardware. Therefore, the BAREFOOT WEDGE-100BF-32X p4 hardware switch has been chosen. It is equipped with 32x100GE interfaces and powered by a TOFINO NPU that can reach 6.4 Tbps. There will be two sites connected by GÉANT equipped with 2xEdgecore Wedge100BF-32X devices (6 x 40G QSFP adapter, 6 x 10G SFP). Currently, the equipment is under procurement by WP7.

Additionally, Jisc and SWITCH will connect to this testbed (see Figure 4.2) with one switch each and RENATER will do the same with two switches, increasing the European testbed to eight machines spread across multiple European NRENs.

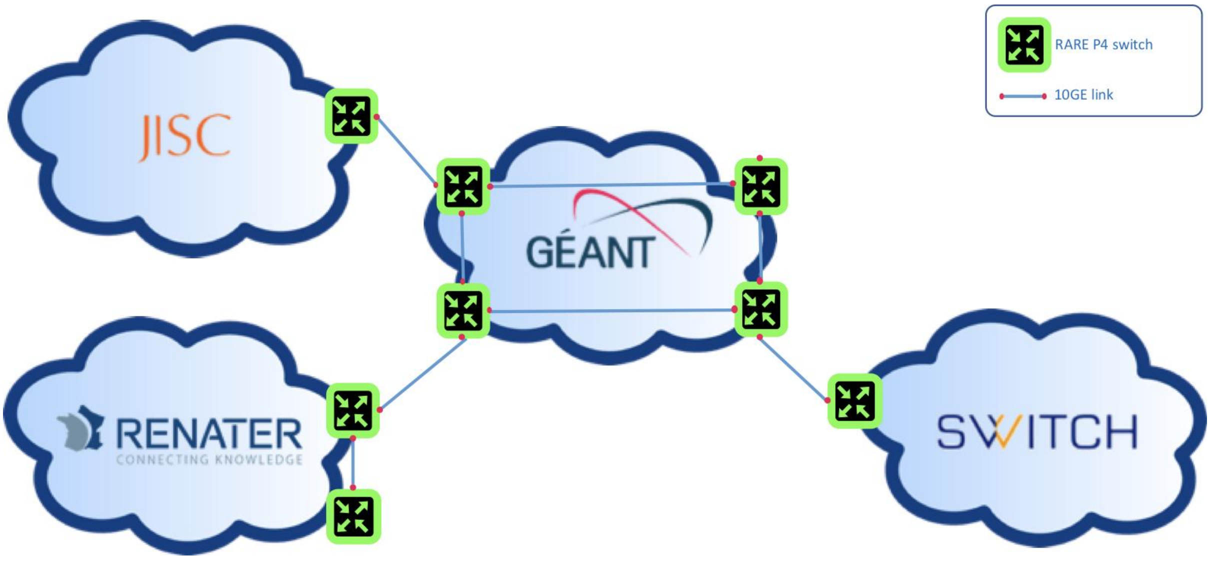


Figure 4.2: GN4-3 WP6 T1 RARE European testbed

4.1 RARE Project Status

WP6 T1 has built a full integration environment for the RARE project. This required that the team established a liaison with the P4lang P4.org organisation and with BAREFOOT. Starting with a pure P4lang environment, a set of packages using the BMv2 model have been created in GitHub as source code modifications to Ubuntu 16.04 and Ubuntu 18.04. But the BAREFOOT switch uses ONL which is inherently a Debian-stable distribution, so a set of P4lang packages have also been built upon for Debian 10 (stable).

The team has also created a RARE GÉANT BitBucket Git, a private git repository, which contains RARE code for the proprietary switch image from the BAREFOOT company. This switch uses a silicon NPU called TOFINO. All the code from the RARE GitHub and the RARE GÉANT Git will get the same structure but in this private Git the target architecture is instead TOFINO bf_switchd.

Several unit labs were created in this development environment to address all features and use cases for the NREN R&E context. Figure 4.3 shows a topology that could be created to test an MPLS P router use case that requires a combination of IPv4 forwarding, IS-IS, and MPLS.

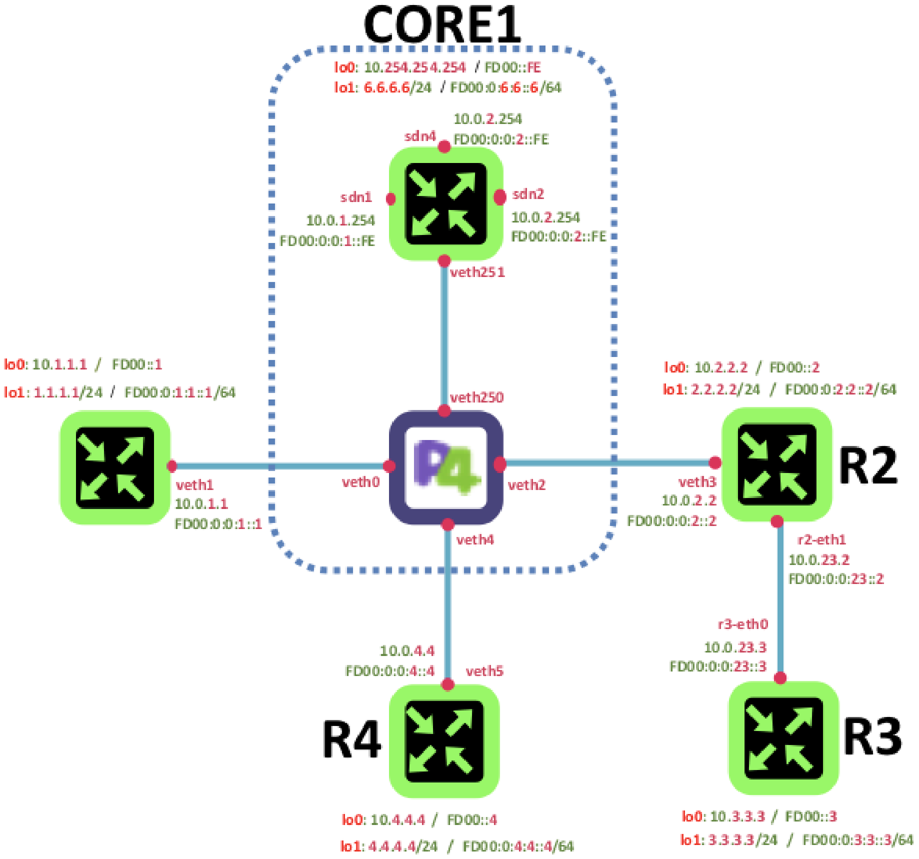


Figure 4.3: Example RARE lab topology

The integration work has made very good progress and at this stage FreeRtR is able to drive a P4 data plane. This breakthrough will allow the WP6 T1 team to implement many use cases, like IPv4, IPv6, MPLS, and SR-MPLS. The next step will be to implement it on the Barefoot Switchd virtual environment, and then to implement it on real P4 hardware.

5 Conclusion

White box hardware and data plane programmability offer the potential for both a technological and economic breakthrough that would be of significant interest to NRENs and R&E networks. There is now the possibility to run different network operating systems (open-source or commercial) over commodity hardware. Moreover, it is now possible to have an open source data plane, and one upon which new features can be rapidly created.

The WP6 T1 work on evaluating white boxes for research and education demonstrates that white boxes can be used as CPE and Internet eXchange point switches. Based on the ongoing work, it is likely that white boxes can also be used for the DC fabric use case.

The management business decision on adopting white box solutions should not only be based on technical considerations and TCO but also on the workload of their team, their capacity to hire staff, the flexibility of the solution, independence from vendors, the strategic plan, etc. For use cases that require more routing features like LER/PE, the NOSs that are available now could, currently, be limited.

WP6 T1 will continue the work on its identified use cases (data centre, GIX, CPE, P/LSR) and will provide a technical analysis once the work is completed.

DDoS algorithms using a novel sketch-based approach have been implemented on a virtual P4 environment and the implementation on P4 hardware is ongoing. The work will require adaptations as some functions used in the virtual environment are not available on the P4 hardware for its implementation on a physical P4 switch.

In-band Network Telemetry (INT) has the potential to allow novel approaches to network monitoring and debugging to be implemented, and can significantly improve network management, using just a few nodes supporting INT. A new implementation on $P4_{16}$ will be provided.

Thanks to the development of unit labs (IPv4, MPLS, Segment Routing etc.), the RARE project has already demonstrated that data plane routing features are feasible, and that the integration of a control plane (FreeRtr) and a P4 data plane is feasible, even if there is a lot of work to be done to finish the full integration. The next step is to integrate this on a virtual environment specific to a chipset (TOFINO) and then to implement it on a hardware P4 switch.

The Deliverable D6.5 *Network Technology Evolution Report* will provide an update on the work reported here in M15 (March 2020) of the project.

References

[Barefoot] https://www.barefootnetworks.com/products/brief-tofino/

[Benycze] https://github.com/benycze/PhD-Thesis

[Buffers] *Optical Packet Buffers for Backbone Internet Routers*, Neda Beheshti, Emily Burmeister, Yashar Ganjali, John E. Bowers, Daniel J. Blumenthal and Nick McKeown

[CESNET] https://p4.org/events/2017-05-09-p4-workshop/

[DPDK] https://www.dpdk.org

[FreeRtr] http://freerouter.nop.hu/

[INT] https://p4.org/assets/INT-current-spec.pdf

[Jim_Warner] https://fr.slideserve.com/kasper-wolfe/speed-match-buffers

[Liberouter] https://www.liberouter.org/combo-200g2ql/

[Merchant_Chips] https://www.nextplatform.com/2018/06/20/a-deep-dive-into-ciscos-use-of-merchant-switch-chips/

[Mininet] http://mininet.org/

[OCP] https://en.wikipedia.org/wiki/Open_Compute_Project

[ONIE] http://onie.org/

[Open_vSwitch] https://www.openvswitch.org/

[Packet_buffers] https://people.ucsc.edu/~warner/buffer.html

[Roy_et_al] *Inside the Social Network's (Datacenter) Network* – Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren

[RFC2889] *Benchmarking Methodology for LAN Switching Devices*, R. Mandeville and J.Perser, IETF RFC 2889, August 2000.

[SFLOW_sampling] https://blog.sflow.com/2009/06/sampling-rates.html

[X86_router] https://hasanmansur.com/2018/06/05/virtual-edge-platform-vep-4600-overview-dell-emc-networking/

[Workshop] https://wiki.geant.org/display/SIGNGN/2nd+SIG-NGN+Meeting
https://eventr.geant.org/events/3050

Glossary

ACL	Access Control List
ASIC	Application-Specific Integrated Circuit
BBR	Bottleneck Bandwidth and Round-trip
BGP	Border Gateway Protocol
BM	Behavioral Model
CapEx	Capital Expenditure
CoPP	Control Plane Policing
CPE	Customer Premises Equipment
CPU	Central Processing Unit
DC	Data Centre
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoH	DNS-over-HTTPS
DPP	Data Plane Programming
EVPN	Ethernet VPN
FPGA	Field Programmable Gate Array
FRR	Free Range Routing
GIX	Global Internet eXchange point
HDL	Hardware Description Language
HPC	High Performance Computing
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IGP	Interior Gateway Protocol
INT	In-band Network Telemetry
IP	INternet Protocol
IS-IS	Intermediate System to Intermediate System
IX	Internet eXchange point
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LHC	Large Hadron Collider
LSST	Large Synoptic Survey Telescope
LPTS	Local Packet Transport Services
LSR	Label Switch Router
MPLS	Multi-Protocol Label Switching
NBD	Next Business Day
NFV	Network Functions Virtualisation

NIC	Network Interface Card
NOS	Network Operating System
NPU	Network Processor
NREN	National Research and Education Network
ONL	Open Network Linux
OOB	Out Of Band
OpEx	Operational Expenditure
OSPF	Open Shortest Path First
P4	Programming Protocol-Independent Packet Processors - programming language
PE	Provider Edge
PISA	Protocol Independent Switch Architecture
PoP	Point of Presence
QoS	Quality of Service
R&E	Research & Education
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RARE	Router for Academia, Research and Education
RTT	Round-Trip delay Time
SDN	Software Defined Networking
SR-MPLS	MPLS Segment Routing
SSH	Secure Shell
T	Task
TACACS	Terminal Access Controller Access-Control System
TACACS+	Terminal Access Controller Access-Control System Plus
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TOR	Top of Rack (switch)
UDP	User Datagram Protocol
VHDL	(VHSIC-HDL) Very High Speed Integrated Circuit Hardware Description Language
VLAN	Virtual LAN
VoIP	Voice over IP
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VRRP	with using the Virtual Router Redundancy Protocol
VXLAN	Virtual Extensible LAN
WP	Work Package