OpenFlow-based Migration and Management of the TouIX IXP

Rémy Lapeyrade[1,2], Marc Bruyère[1,2], and Philippe Owezarski[1,2]

[1]CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
[2]Université de Toulouse, LAAS, F-31400, Toulouse, France

Abstract

The Internet eXchange Points (IXP) are essential for the Internet evolution as they empower high bandwidth low latency and inexpensive local traffic peering as opposed to transit traffic. On the other side, Software Defined Networking, or SDN for short (and its pre-dominant realization, the OpenFlow protocol) enables more dynamic network programmability to control network behaviour via open interfaces, as opposed to the legacy closed-box solutions and proprietary-defined interfaces.

This paper then reports the work that has been done to migrate the TouIX IXP located in Toulouse (France) from a traditional to a full OpenFlow IXP. It especially describes how switches have been selected, configured and installed, and presents the management tools that have been developed, together with performance and service measurements and evaluations. To the best of our knowledge, from may 2015, TouSIX (the SDN evolution of TouIX) is the first Internet Exchange Provider in Europe to fully leverage OpenFlow for its day-to-day operations.

I. Introduction

To face topological, traffic engineering and network service issues of the Internet, Internet eXchange Points (IXP) have been designed and are replacing the global transit model of the Internet. IXPs are fabrics where Internet Service Providers, carriers, content providers and other Internet companies come together to exchange traffic. IXPs are essential for the Internet evolution as they empower high bandwidth low latency and inexpensive local traffic peering as opposed to transit traffic. Today there are already hundreds of IXPs around the world, mainly located in big cities in developed countries. Small and medium-sized cities often still rely on commercial peerings for their local traffic to reach IXPs, even if their physical locations are very close.

On the other side, Software Defined Networking (SDN) enables more dynamic network programmability to control network behaviour via open interfaces, as opposed to the legacy closed-box solutions and proprietary-defined interfaces. We postulate that the divide between the interconnection fabric and the content/service perspective requires a fundamentally different approach to the management of Internet Exchange fabrics. We argue that shifting intelligence from the control plane of current IXP fabrics to their data plane is the key point to improve their scalability, reliability and manageability. Research on SDN technologies and primarily its pre-dominant realisation, the OpenFlow protocol, has then developed a wide range of applications to improve network functionality. OpenFlow control applications can improve network management, monitoring and performance, while being backward-compatible with data plane protocols and end-host network stacks. As a result, within only a few years since the definition of the first version of the protocol, many vendors have introduced production-level support in an effort to transfer innovative research output to the market.

This paper reports the work that has been done to migrate the TouIX IXP located in Toulouse, France from a traditional to a full OpenFlow IXP. TouIX is a non-profit neutral Internet eXchange Point organization founded in 2005. It provides an interconnected network infrastructure at 4 PoPs around the city of Toulouse, and is interconnected to the Paris FranceIX and LyonIX IXPs. TouIX was renamed TouSIX (Toulouse Software Internet eXchange) in late 2014 and, to the best of our knowledge, from may 2015 is the first Internet Exchange Provider in Europe to fully leverage OpenFlow for its day-to-day operations.

The remainder of the paper is as follows: section 2 gives technical details on how the TouIX IXP was built, and lists its related technical and network services issues. Section 3 describes the installation and deployment of the OpenFlow equipments. It especially motivates the selection of Pica8 switches, explains how technical issues have been solved and presents preliminary results. Section 4 then presents the new software management tools that have been designed for TouSIX. Section 5 exhibits a significant set of measurements and evaluation performed on the new TouSIX IXP, exhibiting the benefits of the migration to an SDN based IXP. Section 6 then relates some existing work before concluding the paper in section 7.

II. TouIX: Legacy architecture and raised issues

Figure 1 shows the topology of TouIX. The primary and first site to be located was *Cogent*, where the BGP route server was installed. The Cisco equipment[1] being used in the fabric was configured with three different VLAN tags: admin[2], TouIX_VLAN[3] and France_IX_VLAN[4]. As the Cisco equipment being used was getting older and not appropriate for the fabric anymore, the idea of shaping a completely new Internet exchange to ease the management came to mind.

[1]TouIX used Cisco WS-C3550-12G (TLS00), WS-C3560G (Cogent), WS-C2960G-24TC-L (Zayo) and WS-C2960G-24TC-L (Hotel Telecom)

[2]Used for admin operations on the TouIX network

[3]TouIX DataPath

[4]FranceIX is the larger IXP in France and TouIX allows its members to access France-IX

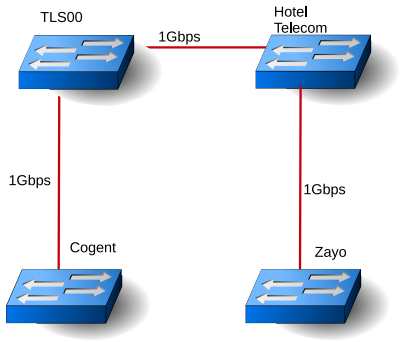

Figure 1. TouIX legacy architecture

The overall management operations of the TouIX IXP fabric represent a complex task that is not easy to perform with existing tools. It was especially hard to analyse and fix several issues. There was for instance the case of several broadcast storms experienced by the TouIX fabric. The lack of an appropriate monitoring infrastructure as well as an irregular appearance of these storm phenomenons made their root causes very hard to identify. The Cisco switches being used did not provide appropriate filtering features to restrict these kinds of undesired traffic that might harm the fabric. There was also a strong requirement for improving the protection of the peering routers against undesired traffic, as well as improving the stability and manageability of the whole infrastructure to prevent the aforementioned phenomenons. Furthermore, the TouIX architecture is suffering from a lack of redundancy of spare equipments.

These various operations issues together with the huge management complexity have motivated the change from the legacy TouIX to TouSIX, in particular :

- Wrong members configurations can create loops and broadcast storm [1], or expose information through discovery protocol like Link Local Discovery Protocol LLDP or Cisco Discovery Protocol CDP. Incorrect IXP switch configurations can also create service disruption.
- The service management complexity requests administration staff members to be familiar with legacy equipment of specific vendors, generally working on a vendor specific Command Line Interface (CLI). A simple web based application will help to simplify all maintenance and management operations.
- TouIX is lacking from a monitoring system able to see the amount of traffic exchanged between members, and to identify types of traffic, for instance.

These operational issues then drove the need of a more generic and easily programmable IXP fabric. TouSIX then aims at introducing an OpenFlow based IXP fabric managed through a web application, and with a fine grained monitoring, for better controlling the network, and more easily managing and operating it.

III. TouSIX architecture

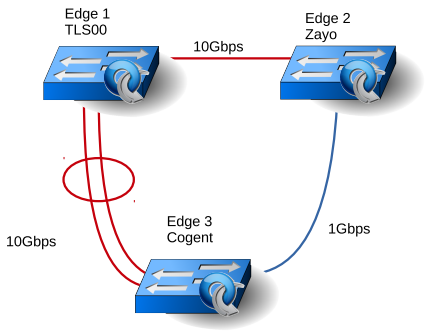

Figure 2. TouSIX architecture

Figure 2 exhibits our final choice for the TouSIX topology.

This new network covers three points of presence, all linked directly to each other. The link between TLS00 and Cogent is an aggregated link. The main purpose of this type of interconnection is to increase the availability between these two switches. This functionality is entirely managed by the virtual switch via LACP, making it invisible for the controller[5]. One link (Cogent-Zayo) is set as a passive link, for privileging the bandwidth usage of the two other links, and then being able to benefit from some redundancy in case of link failure, or simply to switch path of data for adapting traffic engineering policies to traffic evolutions, for instance. This switching capability is implemented using the Fast-Failover capability of the OpenFlow protocol.

For TouSIX, we selected the Pica8 switches to support our architecture. The main motivation for these equipments is the implementation of Open vSwitch [2] on PicOS. It allows us to handle an open-source solution for switching, and facilitate the simulation of the topology in a virtual environment. In the OpenFlow architecture, it is also required for managing the OpenFlow agents of the switches, to install a controller. It has been added on top of the BGP route server, as it is the only available server in our new TouSIX architecture. At this stage of the TouSIX development, the OpenFlow controller is using for communicating with all OpenFlow agents a dedicated VLAN of the ancient (but still running) TouIX architecture. The Ryu controller [3] has been selected for both our OpenFlow rules tests, as well as for the day to day operations.

The key element for migrating to an SDN based IXP relates to the definition of the OpenFlow rules. In our OpenFlow

[5]It is made possible by creating a logical port for the OpenFlow agent of the switch.

installation, we can divide the forwarding rules in two classes. The first one is related to the data plane, and to the forwarding of traffic at layer 2 level, as we made the choice of having a full layer 2 data plane management. Layer 2 forwarding is then achieved by matching directly the destination MAC address with the MAC addresses registered in the TouSIX-manager for the TouSIX users (Ethernet is used as the support technology). The second class of forwarding rules is related to the control plane. They concern specific cases that can be encountered with broadcast traffic as with ARP and NDP [4] (for the IPv6 case) traffic. In that case, the destination MAC address is a broadcast address. To avoid the aforementioned broadcast storms issues, these ARP or NDP packets are not broadcasted on the TouSIX network, but sent to a single user. In that case, the matching of the destination address is considered at the layer 3 level, considering the IP address contained in the ARP (or NDP) request, thus solving the ARP (or NDP) request. All these addresses are recorded in the TouSIX-manager. Recorded MAC addresses in the TouSIX-manager are, for each user, the one of their hardware Ethernet board. On the other side, the related IP addresses are assigned by the TouSIX administrator. Extra OpenFlow rules have been created for implementing this way of resolving ARP or NDP requests that permits fixing the ARP and NDP storm issues.

More generally, all packets not matching the OpenFlow rules are dropped. Thus, unauthorized traffic is dropped at the port entry for all the members.

The application of rules in the switches are taking advantage of the secure mode. This way, if the session with the OpenFlow controller is lost, the rules in production are kept active (otherwise, the switches would have run in the ancient legacy mode for the forwarding. This forwarding relies on the MAC learning principle that is the source of ARP storms). Therefore, the switches are less dependent on the state of an OpenFlow controller, i.e. even in case of failure of the OpenFlow controller, the network switches can continue working in an efficient way. This is as strong improvement for the reliability and liveliness of the IXP.

A. TouSIX architecture liveliness validation

Before going further on the deployment phase, it is first needed to check that this new architecture can bring significant improvements for the IXP management. It especially aims at verifying the liveliness of this new architecture. The first part of this validation deals with testing if the switches are correctly and efficiently handling link failures by testing the Fast-Failover capabilities. Second, all the OpenFlow rules deployed on the switches have been tested to check whether they work or not. All the tests are done by running the testbed depicted on Figure 3.

One host is present on each switch. They generate traffic to all other hosts on the topology. In addition, two routers are present to maintain a BGP session. The list of considered failures is:

1) Link failure
2) Electrical failure of the switches

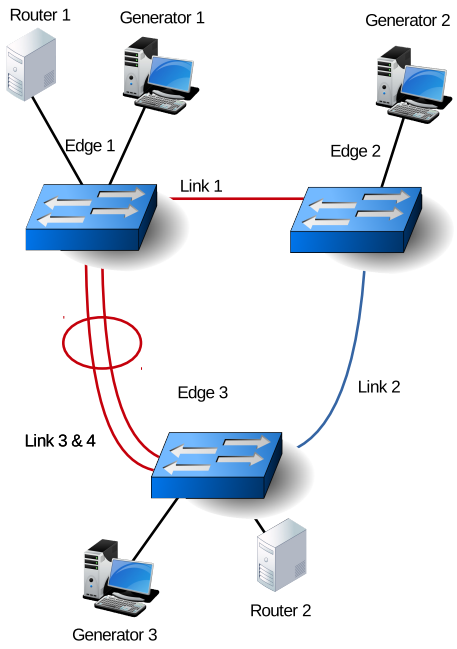

Figure 3. TouSIX test topology

3) Complete shutdown of the network (T_0 test)
4) Shutdown of the OpenFlow controller

The results of these tests were positive. No particular incident has been detected on simulations or on the testbed, as shown on Table I. This table represents all the tests executed. A test is considered as successful if the behaviour during the test of all devices (switches and routers) is as expected for a legacy network in similar cases. In particular, it is checked that the traffic generated by the hosts is forwarded on the correct path. If there is no path available, it is verified that packets of the sending generator are dropped. Similar verifications have been done for the BGP session between Router1 and Router2.

B. Deployment

The deployment of the new TouSIX architecture, changing radically from the legacy TouIX one, is not an easy scenario. Due to the nature of the deployment (that is a complete migration of a commercially running infrastructure), we need to really care about the network behaviour during the migration period. In addition, one of the main attention for this migration is to keep a viable backup solution, for rolling back the network in case the migration fails. It is also essential to maintain the connectivity and quality of service for all connected members during the migration period, including the ones that are not ready to migrate to TouSIX at the date of the migration. This involves keeping both TouIX and TouSIX infrastructures running for several weeks. It is also needed not to impact the traffic sent toward the FranceIX IXP.

As a consequence, we applied some modifications on the TouSIX fully planned services:

- FranceIX VLAN access, which is available on the TouIX

Table I
TOUSIX TEST RESULTS

Test#	Test bed state	Generators	Router2	Router1
0	All links & equipments up	OK	OK	OK
1	Link1 down	OK	OK	OK
2	Link1 up	OK	OK	OK
3	Link2 down	OK	OK	OK
4	Link2 up	OK	OK	OK
5	Link3 down	OK	OK	OK
6	Link3 up	OK	OK	OK
7	Link4 down	OK	OK	OK
8	Link4 up	OK	OK	OK
9	Link3&4 down	OK	OK	OK
10	Link3&4 up	OK	OK	OK
11	Edge1 power Off	OK	OK	OK
12	Edge1 power On	OK	OK	OK
13	Edge2 power Off	OK	OK	OK
14	Edge2 power On	OK	OK	OK
15	Edge3 power Off	OK	OK	OK
16	Edge3 power On	OK	OK	OK
17	Edge1&2&3 power Off	OK	OK	OK
18	Edge1&2&3 power On	OK	OK	OK
19	Ryu Off	OK	OK	OK
20	Ryu back On	OK	OK	OK

network, will not be transferred to the first version of the TouSIX architecture.

- A Layer 2 gateway between TouSIX and TouIX has been added. This is intended to keep the route server and the not-ready-for-migration members be connected to TouSIX members and services.

Figure 4 shows the new TouSIX topology at this partial level of the migration. The layer2 gateway is located on Edge3, connected to the two routers which can then communicate with any other TouSIX routers. Moreover, the passive link is not available at this stage of the migration (for simplifying the process).

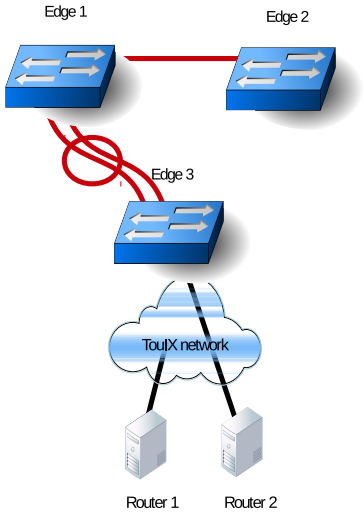

Figure 4. TouSIX Migration topology

Similarly, the aggregated link was to difficult to migrate at this first stage of the OpenFlow deployment; activating this aggregated link was working perfectly on simulation, but not on the production architecture. It seems that there is an inconsistency between simulation and real world with the current PicOS implementation. Up to now, this link activation has not been made possible on the production network. Fixing this issue is one of the objectives for migration to a second TouSIX architecture.

IV. THE TOUSIX-MANAGER

The new TouSIX IXP being set-up, an efficient and convenient management tool of the control plane is required. For convenience, this tool, called TouSIX-Manager, has been designed and developed as a web platform. Its main objectives are:

- Simplifying the maintenance of the TouSIX IXP.
- Making easy the integration of new services through the OpenFlow protocol.

One of the requirements for the TouSIX-Manager is to be independent of any specific controller. The functionalities of the TouSIX-Manager controller consist of:

- Controlling the OpenFlow rules needed for a good functioning of the network. For instance, in case of a switch software or hardware reboot, the TouSIX-Manager must restore a stable state of the OpenFlow rules. It can be defined on a database or a local file.
- Performing OpenFlow operations without passing by any software abstraction layer.
- Sending periodically some pre-defined informations from the network to a database. These informations will be used for raising OpenFlow alerts or display statistics.

This was achieved by using the Ryu controller. As a consequence, our main architecture is divided into two main components: One or several controllers, and web applications bundled to create a fully functional website. The communication between the website and the controller uses HTTP requests.

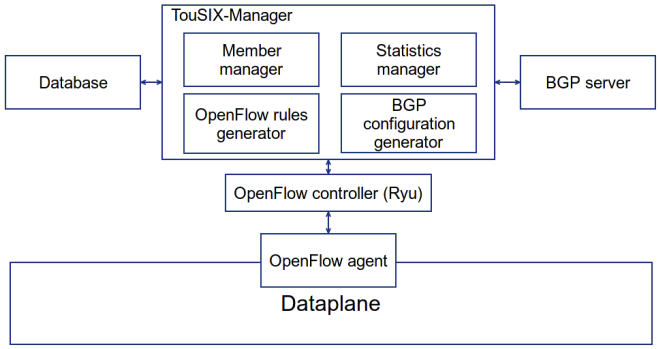

Figure 5. Architecture of the TouSIX-Manager.

On Figure 5, the website designed for managing the new TouSIX IXP is divided into four main components. The first component aims at generating all the OpenFlow rules needed for the exchange point. For any topology defined and stored on the database, all or per-switch rules can be generated on

request. It is also possible to generate a certain set of rules, which correspond to functionalities we want to implement in our topology. The result is then stored in JSON format [5] in the database. On another side, this component aims at managing the OpenFlow rules into the controller (Ryu in our case).

The second functionality of the website is to store information related to OpenFlow counters. By matching certain fields on an OpenFlow rule, without any forwarding action, we can retrieve the number of packets matching this rule. With a periodic monitoring of these counters, we can establish the bandwidth used by the matching packets. Thus, it is possible to aggregate values for representing global traffic, like in Figure 6.

On the current version of the TouSIX-Manager, per-member statistics on IPv4, IPv6, ICMPv6 and ARP activities are provided. More statistics can of course easily be monitored according to specific operator requirements.

The third component manages all the members and users created on the website. The behaviour of this module is what we could expect from a network equipment management tool. It is not as exhaustive as other specialized solutions [6]. But, as an improvement, the process for adding a new member to the IXP is automated. The administrator of the IXP only needs to validate data sent by the new member, and assign one access port. But there is no more technical intervention needed. Thus, it reduces the human error induced by the member or the IXP administrator.

The last functionality is an automatic configuration of our BGP server. If some changes happen on the TouIX network (eg. a new member wants to get connected), it can generate a convenient configuration for the BGP server. This component, with respect to the OpenFlow rules deployment, allows the IXP administrator to create a fully automated procedure for managing the exchange point when adding/removing members.

V. Evaluation

This section addresses two sets of performance measurements: one for assessing the benefit of an SDN based IXP on the management point of view, and the other for evaluating the statistic manager performance.

a) Rules generation profiling: At this point of the TouSIX development, we want to know how fast the generation of OpenFlow rules can perform. Specifically, we need to measure the time the web application takes when some procedures (adding a member on the topology for example) are called. The time required for deploying rules on the TouSIX topology[6] will not be evaluated in the context of this paper, focusing only on the duration of TOUSIX-manager web procedure calls. For this purpose, some procedures of our web application have been tested with the native python profiling module [7].

Figure 7 depicts the total duration for generating all possible rules. As a result, the generation of rules takes much less than one second. Note that for the day-to-day operation of the TouSIX IXP, around 500 OpenFlow rules are required. With the current TouSIX configuration, it is not possible to have more realistic rules, and then push further the rule generation procedure evaluation. Figure 7 also exhibits an almost linear time (at least on its main part) for computing all the rules necessary for running the TouSIX IXP. It just exhibits that because of initialization time, the generation of the first rules is impacted by limited additional delay. For more than 500 OpenFlow rules, it seems that the rules generation procedure could be reaching its scalability limits. However, as it is not possible to extend this evaluation with more than 500+ realistic rules on the current TouSIX IXP, it is not possible to give a clear conclusion yet. However, by analyzing the tests results, it is identified that most of the call duration are related to database operations. Database operations represent 95% of the total call time. We are currently working on solutions for reducing the time spent waiting a response from the database. The increase of the time for generating more than 500 rules seems to be directly related to this longer and longer database access time.

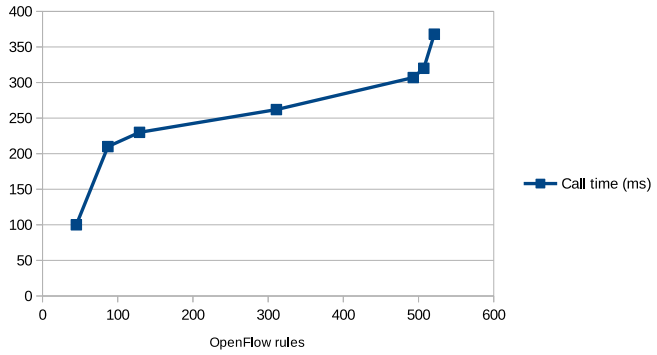

Figure 7. Rules Generation module execution time

b) Statistic manager performance: The statistics manager application is a custom implementation including two main functionalities: the management of a time-series database, and a graphing tool to visualize this database. In the following, the performance of the statistic manager are presented. They have been obtained by stressing it using the Locust [8] stress test framework.

The test performed consists in measuring the time for getting and displaying the results of different functions provided by the statistic manager including, for instance, the times required for posting or getting flow statistics, aggregated traffic statistics, etc. The results have been obtained on sequences of 20 requests per seconds in order to really stress the statistics manager, and also for limiting measurement errors. The results are displayed on Figure 8. It especially exhibits that on an average of 20 requests per second, the mean response time is increasing. It also shows that with the most time consuming requests in terms of computing resources (when computing global traffic statistics for instance), it can take up to 2.6

[6]The deployment of rules that consists in sending and installing OpenFlow rules on OpenFlow agents is performed right after their global generation

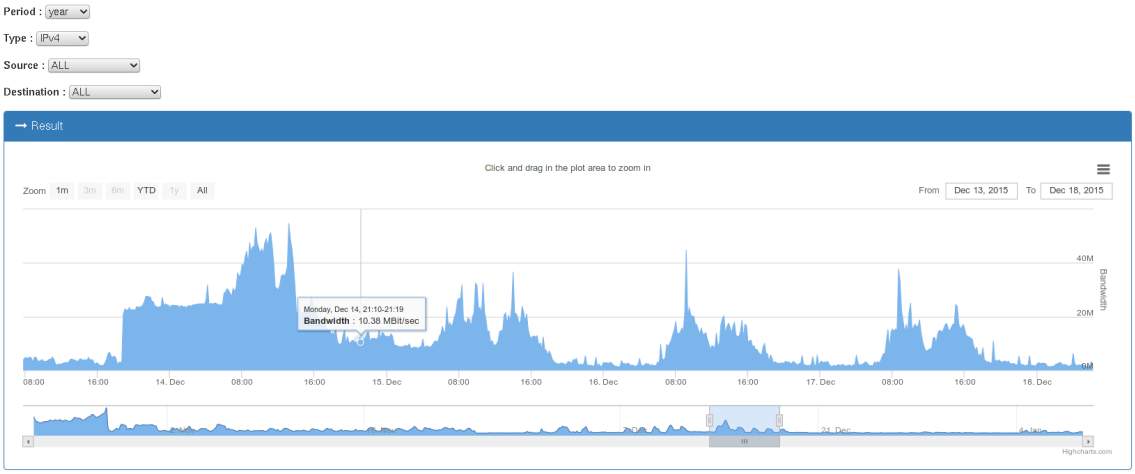

Figure 6. Example of TouIX IPv4 traffic measurement provided by the TouSIX-Manager.

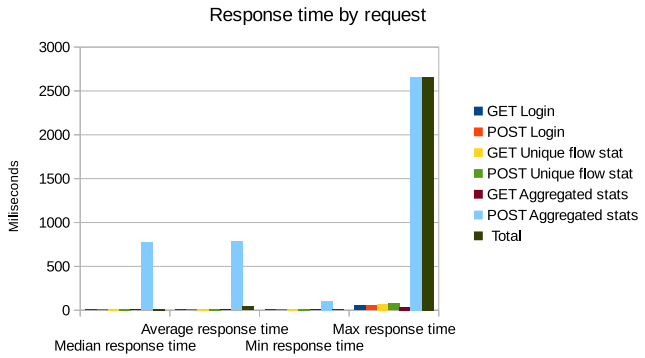

Figure 8. Statistics manager response time

seconds to get the results. In addition, it is observed that these huge delays accumulate, leading to non real-time display of the traffic statistics. This is of course a strong limit of the current version of the statistic manager that will be addressed in future work.

VI. Related works

To the best of our knowledge, only one similar example to TouSIX exists, despite the large interest SDN is raising among the IXP operators community. This is the Cardigan project. Stringer et al. [9], with the Cardigan project, implement a hardware based distributed routing platform using Openflow with RouteFlow [10]. Cardigan is a Layer3 peering fabric that can interfere with the BGP policies of the member. The TouSIX IXP fabric, as the difference, is a layer2 fabric (the does not interfere anyway with layer3) where IXP members can manage their BGP policies. The Cardigan IXP has been in operation for a couple of years from the beginning of 2012 to the end of November 2014. Due to a change New-Zealand local law, Cardigan could not be maintained in operation.

VII. Conclusion

This paper illustrated how an IXP can benefit from an SDN technology as OpenFlow for enhancing reliability and manageability. This has been performed on the real TouSIX IXP where OpenFlow based devices, tools and applications have been successfully deployed, thus demonstrating the practical applicability and benefits of such a solution. As a result, TouSIX now fully leverages OpenFlow for its day-to-day operations. As future work, we target solutions for improving the scalability of the SDN/OpenFlow solution, as well as introducing ways for securing such an SDN based IXP.

References

[1] "Franceix outage website page," https://www.franceix.net/en/events-and-news/news/franceix-outage-notification/, [Online; accessed 03-Jan-2016].
[2] "Open vswitch website," http://openvswitch.org/, [Online; accessed 23-Sep-2015].
[3] "Ryu website," http://osrg.github.io/ryu/, [Online; accessed 23-Sep-2015].
[4] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor discovery for ip version 6 (ipv6)," IETF, Tech. Rep., 2007. [Online]. Available: http://tools.ietf.org/html/rfc4861
[5] "Ryu rest api definition," http://ryu.readthedocs.org/en/latest/app/ofctl_rest.html, [Online; accessed 23-Sep-2015].
[6] "Ixp-manager source code," https://github.com/inex/IXP-Manager, [Online; accessed 23-Sep-2015].
[7] "Python cprofile module," https://docs.python.org/3.5/library/profile.html#module-cProfile, [Online; accessed 17-Dec-2015].
[8] "Locust.io website," http://locust.io/, [Online; accessed 23-Sep-2015].
[9] J. Stringer, D. Pemberton, Q. Fu, C. Lorier, R. Nelson, J. Bailey, C. Correa, and C. Esteve Rothemberg, "Cardigan: Sdn distributed routing fabric going live at an internet exchange," in *Symposium on Computers and Communications (ISCC)*. IEEE, 2014.
[10] "Routeflow website," https://sites.google.com/site/routeflow/home, [Online; accessed 03-Jan-2016].