CityFlow: OpenFlow City Experiment – Linking Infrastructure and Applications

Adam Carter[1], Donal Morris[2], Sachin Sharma[3], Luís Cordeiro[4], Ricardo Figueiredo[2], João Gonçalves[4], David Palma[4], Nick Johnson[1], Dimitri Staessens[3]

[1]EPCC, The University of Edinburgh, UK, [2]RedZinc, Ireland,
[3]Ghent University – iMinds, Belgium, [4]OneSource, Portugal
Email: [1]{Adam.Carter, Nick.Johnson}@ed.ac.uk, [2]{dmorris, ricardo}@redzinc.net
[3]{Sachin.Sharma, Dimitri.Staessens}@intec.ugent.be, [4]{cordeiro, joagonca, palma}@onesource.pt,

***Abstract*— CityFlow is an EU FP7 project, aiming to create a set of multi-autonomous-system OpenFlow experiments on the OFELIA infrastructure to emulate a city of one million inhabitants. In this demo, we demonstrate all of the key components of the CityFlow experimentation stack working together.**

Keywords- Next generation networking, Quality of service, Routing protocols, Soft switching

I. INTRODUCTION

In this demo, we present some results for an OpenFlow Experiment to explore and OpenFlow control plane with a large number of dynamic invocations. The CityFlow project [1] has built an emulation of a control plane of a city of one million inhabitants and defined a number of networks within that city – so called “FlowVille”.

II. FLOWVILLE EMULATION

For FlowVille, we have analysed the network infrastructure for Brussels, population 1.1 million, in order to obtain a reference scenario, which can be emulated by CityFlow on the OFELIA testbed in Ghent, the Virtual Wall. Figure 1 shows the reference scenario, consisting of 3 users (for emulating one million end-users) in the access network, 3 ring of OpenFlow switches in the aggregation network, 1 ring in the core network, and 1 ring the CDN network. All these rings form multiple autonomous system scenarios for our experiments. For emulation, we assume that each autonomous system is controlled by a single controller.

Our experiment is based on an control plane stack consisting of: (i) a Virtual Path Slice (VPS) engine [2] which manages concatenating bandwidth slices in multiple autonomous systems, (ii) FloodLight [3] which is an openflow controller, (iii) RouteFlow [4] which is a mechanism to integrate interdomain BGP forwarding advertisements from adjacent autonomous systems into OpenFlow rules, (iv) QueuePusher which is a mechanism for installing queues on an Open vSwitch [5] (v) OpenvSwitch, which is an OpenFlow softswitch. We stimulate the control plane stack with high volume of requests from a test harness which functions as a pulse generator sending requests to the northbound API of the VPS engine. The stack is illustrated in Figure 2.

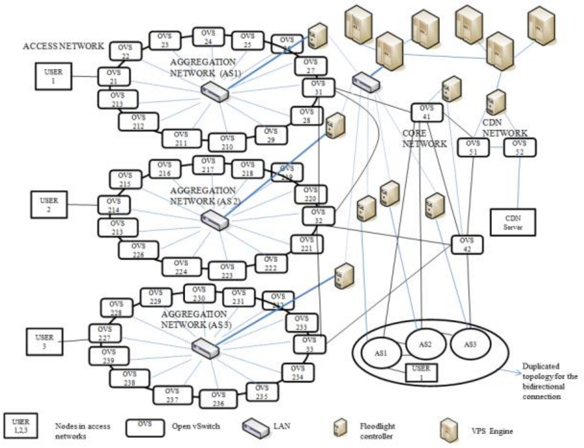

Figure 1. FlowVille emulation experiment on the OFELIA testbed

In the demo we show the pulse arrival, processing interval of a system event and the volume of connection arrivals. Our purpose is to maximise the busy hour capacity of the system for arriving connection establishment events.

The results that we will demo directly will show the full experiment stack set up and running on a test bed on the site of one of our project partners (RedZinc). This test bed has also been used to undertake experiments with other OFELIA islands. A screenshot taken from one of the demo applications is shown in Figure 3. This display is presented in a web browser window and shows the current performance of the system in real time.

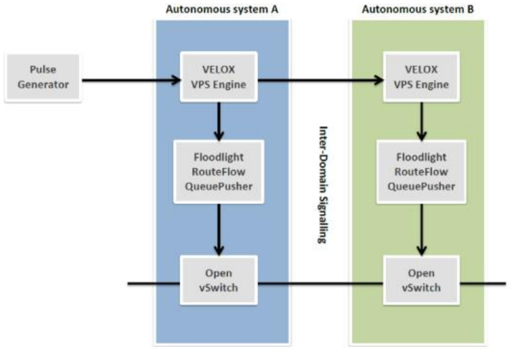

Figure 2. Emulation framework and software stack

We will also summarise our results from a series of experiments that aim to:

1. Undertake a system test and demonstrate Virtual Path Slice Signalling and OpenFlow interworking
2. Undertake a stress test of the stack using the large-scale deployment on the Virtual Wall at iMinds
3. Understand the response to failures in the system, both in the network and in the VPS engine itself.
4. Explore how the components of the stack operate when deployed across multiple OFELIA islands, to learn about real-world deployment issues.

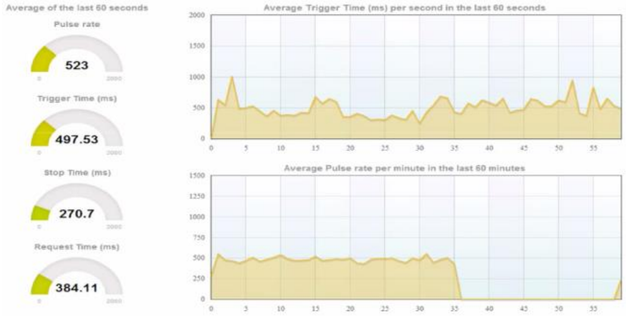

Figure 3. A screenshot from the demo showing results from a control plane stressing experiment in real time

III. Key Findings

From the system test experiments we learned that using the stack that we had integrated, the VPS bandwidth guarantee could be set-up across multiple domains. We conclude that it is possible to integrate the following OpenFlow technologies in a combined stack: Floodlight for OpenFlow switch control; RouteFlow for propagation of routes between different autonomous systems using IPv4 and BGP; Open vSwitch; Virtual Path Slice Engine, a multi provider signalling system for coordinating bandwidth slices between different autonomous systems and a Queue Pusher module for realising slicing in the data plane of the Open vSwitch using rates applied to the queues for a particular DiffServ code point.

We could conclude that the East-West aspect of the data plane giving end-to-end connectivity over a mixed IP and multi-autonomous OpenFlow network is operational. In addition we concluded that the North-South aspect of the control plane communicating from the application layer (as manifested by the pulse generator) to the queue in the data plane is functional.

From the stress test experiments we conclude that the VPS Engine can scale to a high volume of flow invocations and terminations, to support a busy hour flow invocation capacity of 75000 events on mid range servers. This aligns with what might be expected in an OpenFlow area. We observed that we conclude that Floodlight suffers from bottlenecks in the flow installation process, which should be resolved for a high performance environment.

We conclude that the VPS Engine can slice bandwidth in an integrated OpenFlow environment and coordinate that between multiple autonomous systems (e.g. CDN network to a customer's access network).

From the response to failure experiment we learned that high quality of service can be achieved for business customers even on failure conditions using our framework [6]. We learned that if the enough bandwidth is present in the restoration path, neither business customers nor best-effort customers would receive degraded service. In addition, if there is a limited bandwidth available in the restoration path, the best-effort customers will first receive the degraded service. The business customers only receive the degraded service when there is no enough bandwidth available in the restoration path to accommodate all the business customers' traffic. In this scenario, there will not be any interference between the business customers and best-effort customers.

From the multi-island experiment we conclude that the OFELIA test bed can support multiple islands conducting experiments. In our experiment we used the OFELIA Virtual Wall segment at high scale and multiple OFELIA islands at low scale to validate interoperability. A signalling experiment can be conducted on mixed OFELIA OpenFlow islands and external OpenFlow islands.

Acknowledgment

This work has received funding from the EU FP7 under agreement number 317576 (CityFlow).

References

[1] CityFlow: www.cityflow.eu

[2] T. Braun, Diaz, M., Gabeiras, J.E., Staub, End-to-End Quality of Service Over Heterogeneous Networks, Springer, 2008

[3] Floodlight Controller: http://www.projectfloodlight.org/floodlight

[4] Esteve Rothenberg et al. Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking, HotSDN, 2012.

[5] Open vSwitch: www.openvswitch.org

[6] S. Sharma, D. Staessens, D. Colle, D. Palma, J. Goncalves, M. Pickavet, L. Cordeiro, and P. Demeester, Demonstrating Resilient Quality of Service in Software Defined Networking, IEEE INFOCOM, pp. 133-134, 2014