Cardigan: Deploying a Distributed Routing Fabric

Jonathan Philip Stringer, Qiang Fu
Victoria University of Wellington, New Zealand
joe@wand.net.nz
qiang.fu@ecs.vuw.ac.nz

Christopher Lorier, Richard Nelson
University of Waikato
Hamilton, New Zealand
cml16@waikato.ac.nz
richardn@waikato.ac.nz

Christian Esteve Rothenberg
University of Campinas (UNICAMP), Brazil
chesteve@dca.fee.unicamp.br

ABSTRACT

Hybrid Software-Defined Networking (SDN) systems are an active area for network research, with many organisations exploring the opportunities unlocked by the de-coupling of network control from packet forwarding. Previous work has suggested that a hybrid networking model will pave the way for migration towards SDN, through interoperability with legacy devices. However, questions remain over the operation of such systems in production environments. In order to explore the challenges of hybrid SDN systems and build operational confidence, we built a simple distributed router using OpenFlow and deployed it at a public Internet exchange. This implementation provides insights into the challenges involved with using these technologies, and suggests the viability of mixed device environments despite the limitations of early OpenFlow implementations.

Categories and Subject Descriptors

C.2.6 [**Internetworking**]: Routers

Keywords

OpenFlow; Network Virtualization; Distributed Router

1. INTRODUCTION

SDN and OpenFlow [1] have given us many new tools for reimagining our approach to layer 3 networking. However, these technologies are in their infancy, and their unproven nature and misconceptions caused by a lack of familiarity have meant these technologies have seen little use in production. Cardigan is a project to generate confidence in SDN by deploying an OpenFlow networking environment in a production setting. This deployment will also help to identify practical issues with the roll out of SDN environments, detecting any incompatibilities with legacy networking devices and protocols, and finding clues as to possible implementation barriers for future wider deployments.

SIGCOMM'13, August 12–16, 2013, Hong Kong, China.
ACM 978-1-4503-2056-6/13/08.

To deploy a new technology in existing environments, it needs to provide a migration path from existing systems, and there must be motivating reasons for adoption. While the commoditisation of network hardware is expected to drive costs down, this alone may not provide enough benefit to warrant the replacement of existing hardware. The Cardigan project is positioned as a crucial first step in showing the viability of deploying SDN hardware side-by-side with existing production equipment, and seeks to assess the advantages that SDN can offer the WAN.

We created a distributed router based on the RouteFlow [2] design, which allows a mesh of OpenFlow switches to be aggregated and represented as a single logical router. This simplified network structure provides benefits to operators in terms of lowered maintenance time and reduced likelihood of misconfiguration. Ongoing benefits are also provided through the ease of modification and simplified diagnosis of problems.

2. CARDIGAN DESIGN

The Cardigan design aggregates a set of flow forwarding devices (datapaths), and represents them logically as a single router. Each datapath may be connected to external peers (via external links), and should be connected to other datapaths in the distributed router (via inter-switch links).

Each of these datapaths sets up an OpenFlow connection with the RouteFlow RFProxy application. A topology model is then used to associate ports on the datapaths to ports on a single Virtual Machine (VM) or container in a 1:1 configuration. The association is implemented by connecting the VM to one or more virtual datapaths, each of which creates an OpenFlow connection with the RFProxy application. Finally, the RFProxy application facilitates the forwarding of control plane traffic (e.g. ARP, ICMP, BGP) between external peers and the virtualized routing engines, and translates higher-level route information into OpenFlow flow modification commands.

Furthermore, we introduce a hierarchy of rules based upon a proactive flow installation approach. The highest priority rules provide high-level blocking of entire classes of traffic. For instance, packets which do not contain the appropriate layer 2 address will be immediately blocked. The next priority of rules is for control traffic, which must be destined for the controller IP and be explicitly allowed to be passed to the control plane. Following these, there is a set of priorities for Hosts and Routes. These are sorted in order of prefix to implement longest prefix matching; longer prefixes adopt a higher priority in the rule table than shorter prefixes. Fi-

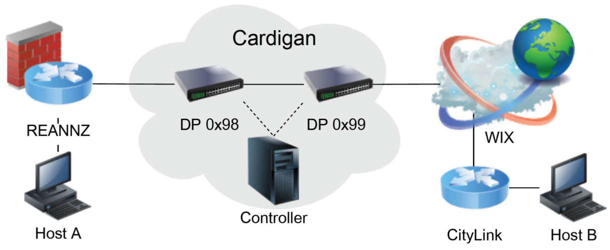

Figure 1: Cardigan pilot deployment.

nally, any traffic that does not match the prior groups will be dropped by default, and not sent to the controller. This reduces attack vectors to the controller, as non-matching traffic does not enter the control plane.

Datapath aggregation is managed with static configuration in RFServer specifying each inter-switch link and its connected datapaths. The initial implementation requires datapaths to be linked in a full mesh –like router line cards and fabric cross-connects– connected to controllers communicating to the same RFServer instance. RFServer then handles forwarding communication to higher-level services for calculating route modification information for those datapaths.

3. DEPLOYMENT

Cardigan deployment consists of a RouteFlow distributed router, connecting the Research and Education Advanced Network of New Zealand (REANNZ) to the Wellington Internet Exchange (WIX). RouteFlow controls two OpenFlow switches, connected by a dark fibre link, one situated at the border of each network (see Fig. 1). The switches used were a Pronto 3290 at WIX and a Pronto 3780 with 1G SFPs at REANNZ, each using PicOS 1.6. The controller was deployed at a third location, connected via an out-of-band layer 2 VLAN. BGP peer sessions were established with a router running within REANNZ and all WIX participants. Routes to the REANNZ network were advertised onto the WIX and traffic was forwarded through the two switches.

Cardigan has been deployed in production for over four months, forwarding customer traffic and sharing routes with ninety-seven other participants of WIX without major incident. As of the time of writing there are 1134 flows on each switch, broken down as follows:

- 8 flows tunneling control plane traffic;
- 98 flows describing directly connected hosts, at the WIX and at REANNZ;
- 1028 flows representing layer 3 routes learnt from peers;
- 1 low-priority flow to drop traffic by default.

The boot time for Cardigan is around one minute, including flow installation. The major bottlenecks for this are surrounding bootup of the NoSQL database/IPC backend and resolution of next-hop MAC addresses for routes. Initial throughput measurements were performed using Iperf between hosts connected to REANNZ and CityLink as indicated in Figure 1. These showed modest TCP performance across the path in the order of ≈800Mbps, peaking at 855Mbps. Given the live nature of the deployment, deeper performance analysis was not conducted. Ongoing updates from the Internet eXchange Point (IXP) provide 3-4 updates every ten seconds, due to ARP timeouts and link changes.

4. CHALLENGES

The initial RouteFlow approach to implementing route propagation from the routing engine to OpenFlow datapaths involved a tight coupling of gateway resolution and flow installation. This works for the common case where a peer will share routes that are accessible via itself. However, in an IXP environment, it is not uncommon for the exchange to provide route servers which advertise routes, but do not forward traffic.

The tight coupling of these functions was observed to cause delays and corruption when installing flows in the cardigan environment. Although these could be alleviated through the use of non-blocking sockets and by increasing Netlink socket buffer sizes, the approach was still insufficient in some circumstances. A revised approach that separates gateway resolution was able to correctly handle routes with reachable gateways, and those which were not reachable would be queued until they are reachable. With these changes, we were able to decrease datapath convergence time by two orders of magnitude.

5. CONCLUSIONS AND FUTURE WORK

We presented design extensions to RouteFlow which allow the aggregation of multiple datapaths, representing them logically as a single device. Furthermore, we implemented this design and contributed the source code back to the public repository [3]. Despite initial limitations, the Cardigan deployment is successfully passing production traffic between REANNZ and WIX. We anticipate further challenges when attempting to scale to larger networks—appropriate monitoring of network resource usage; load-balancing; closest exit usage and effective policy enforcement (c.f. [4]). These topics are part of ongoing efforts to improve the viability of RouteFlow in production environments.

Acknowledgements

The authors are thankful to REANNZ and CityLink NZ for their assistance in the deployment; and to Dean Pemberton, Dylan Hall and Josh Bailey for their continued contributions in the project. This work was done while Christian was employed by CPqD and was partially supported by RNP.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM CCR*, pp. 441–442, March 2008.

[2] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Corrêa, S. C. de Lucena, and R. Raszuk, "Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking," in *HotSDN '12*, Aug. 2012.

[3] The RouteFlow Community, "RouteFlow source code." `http://github.com/CPqD/RouteFlow`, 2013.

[4] N. Feamster, J. Rexford, S. Shenker, R. Clark, R. Hutchins, D. Levin, and J. Bailey, "SDX: A Software-Defined Internet Exchange," *ONS '13 Research Track*, Apr. 2013.