

Revisiting IP Routing Control Platforms with OpenFlow-based Software-Defined Networks

Christian Rothenberg¹, Marcelo Nascimento¹, Marcos Salvador¹,
Carlos Corrêa², Sidney Lucena², Allan Vidal^{1,3} and Fábio Verdi³

¹Telecomm. Research and Development Center (CPqD),
Campinas, SP, Brazil – esteve@cpqd.com.br

²Federal University of the Rio de Janeiro State (UniRio),
Rio de Janeiro, RJ, Brazil – carlos.correa@uniriotec.br

³Federal University of São Carlos (UFSCar),
Sorocaba, SP, Brazil – verdi@ufscar.br

***Abstract.** This work discusses how previous IP routing solutions, like RCPs, are reshaped in light of SDN's clean separation of functions. To materialize this view, we define an initial Routing-as-a-Service platform based on the RouteFlow architecture and its ability to virtualize IP routing operations. To validate feasibility of running novel forwarding logic on top of it, a proof-of-concept service was built where a single process executing BGP is used to map and control all routing interfaces of an entire AS. The obtained results suggest the viability of executing a single, consistent AS-wide business policy that lowers administrative overhead while opening the door to innovative IP-centric services.*

1. Introduction

Providing a clean separation of functions is paramount to any system or network architecture. In the switching/routing networking realm, this is being touted as Software-Defined Networking (SDN) and is characterized by new abstractions resulting from decoupling data-plane forwarding from control and management plane functions. OpenFlow is arising as the protocol of choice between controller and forwarding elements.

Examples of different SDN use cases are increasingly popping up. Curiously, little attention has been paid to the implications of SDN meeting traditional IP routing and forwarding. Remarkably, SDN-enabled APIs provide a new chance for deployment of (past) research, and hence, an opportunity to reheat topics around IP routing (e.g. BGP).

Around one decade ago, research on so-called Routing Control Platforms (RCP) [Ramjee and et al. 2006, Feamster et al. 2004, Van der Merwe et al. 2006, Wang et al. 2009] pioneered refactoring the IP routing architecture to create a logically centralized control plane separated from forwarding elements to focus on the BGP decision process and the route control needs from a large operator's perspective.

In this paper, we re-examine the concept of RCP with the visibility (network-wide, multi-layer, flow and topology map) and direct control capabilities (actual FIB installation, rich matching and instruction set) of the SDN abstraction set and the specifics of the OpenFlow choice. Building upon our earlier work on RouteFlow [Nascimento et al. 2011], we show that a RouteFlow Control Platform (RFCP) can

be used to implement advanced RCP applications by acting as a new indirection layer for routing messages and RIB-(to-FIB)-to-OpenFlow transformations. Finally, we describe experiments of a prototype implementation of a scale-out aggregated BGP router.

2. Background

In essence, RCPs are based on three architectural principles [Ramjee and et al. 2006]: i) path computation based on a consistent view of network state, ii) controlled interactions between routing protocol layers, and iii) expressive specification of routing policies. In contrast to the complex distribution of BGP configuration across many routers, RCPs allow an individual AS to easily deploy new, customer-facing (and revenue-generating) services [Van der Merwe et al. 2006].

3. RFCP Design

The current RouteFlow Control Platform (RFCP) architecture has evolved from the previous prototype designs [Nascimento et al. 2011] to a better layered, distributed system design, flexible enough to accommodate different virtualization use cases. It follows a modular architecture which segregates implementation into three components (cf. Fig. 1(a)):

- **RF-Slave:** Collects routing and forwarding information from the Linux host, where it runs as a user-space daemon. Optionally, for extraction of additional routing information, it hooks into or peers (e.g. iBGP) with the routing engine.
- **RF-Server:** Standalone application responsible for the core logic of the system (e.g., event processing, mapping VMs to switches, resource mgm.). RFCP Services are implemented as operator-tailored modules which use the knowledge information base to implement arbitrary, high-level routing logics (e.g., router consolidation, load-balancing).
- **RF-Controller:** Application on top of any OpenFlow Controller which services the RFCP through switch interaction and network's topology state collection.

In line with the design rationale and best practices of scale-out cloud applications, a programmer-friendly, automatically clustered, redundant datastore centralizes (i) the network view, (ii) the RFCP core state (critical for resiliency), and (iii) any knowledge information base (e.g., traffic histograms/forecasts, flow monitoring feedback, any administrative policies, etc.) used to develop advanced routing applications. In addition, the NoSQL database of choice (e.g., MongoDB, Redis, Cassandra) is used as the pubsub-like message queuing IPC that loosely couples the modules via an extensible JSON-based implementation of the RouteFlow protocol. Without sacrificing performance, the datastore-centric approach to the IPC eases fault-management, debugging, and monitoring.

All in all, RFCP comes as a software artifact capable of instructing the SDN data plane with regard to IP traffic forwarding and which builds over the following premises:

- Ability to define desired routing criteria (i.e. business policy) in an abstraction level which does not require individual configuration of multiple routing processes;
- A unified database that conveys information about the network's physical connectivity state and the logical counterparts, allowing for provisioning and traffic-engineering control logic and services;

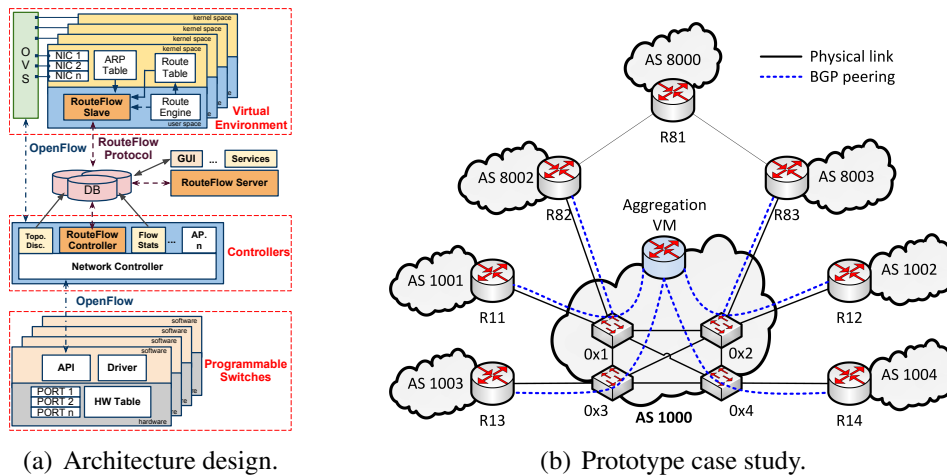


Figure 1. RouteFlow Control Platform.

- Flexible and scalable virtual control plane elements, which provide strong automation and flexibility for in-memory representation of intended control scenarios.

4. Use Cases

Related work has shown the benefits of centralized route control for a more flexible routing [Wang et al. 2009] and connectivity management tasks [Van der Merwe et al. 2006].¹ Innovative use cases include:

- **Traffic Engineering.** A centralized server knows all available (eBGP) paths and hence allows advanced load-balancing (measured- or dollar cost-based), IP route protection schemes, or application-aware performance routing using OpenFlow matching to push traffic into traffic-engineered paths or VRFs.

- **BGP Prefix Independent Convergence (PIC).** OpenFlow group tables (or just multiple-tables) introduced in v1.1 allow implementing Generalized Hierarchical FIB [Filsfils et al. 2011], drastically reducing the update times of next hop actions (e.g., MAC address, label, port-out). Moving away from dynamic route computation by decoupling failure recovery from path computation [Caesar et al. 2010] becomes feasible.

- **Optimal best path reflection.** A common problem of route-reflectors is that it may not yield the same route selection result as that of a full iBGP mesh. With full route visibility in the RFCP, route selection can be done from each datapath's perspective.

- **Simplified customer multi-homing and IPv6 migration.** Enterprise networks suffer from deploying effective multi-homing solutions. RFCP could be used as an upstream affinity mechanism, only sending packets sourced from ISP1 networks via ISP1 (and 2 via 2) by OpenFlow rule matching based on source IP subnet. Similarly, edge migration to IPv6 could be accelerated and cheaply rolled out with OpenFlow match plus encapsulate actions. It might even enable new services that leverage the strengths of IPv6 and mitigate its current weaknesses, especially in the control plane.

¹Not requiring changes to the routers or protocols, use cases from previous work (e.g. planned maintenance dry-outs [Van der Merwe et al. 2006]) can be directly applied to the proposed architecture.

5. Prototype Implementation

Reviewing the topology characteristics that impact routing with the RFCP, the dataplane forwarding domain can be described by datapath IDs and links. The latter can be separated into “trunk” (datapath-to-datapath) and “edge” (datapath-to-device) categories. The RFCP maps every relevant network connections into a memory model.

To verify the platform’s ability to supplement forwarding decisions through RF Services, an actual routing service, called “RFagg”, was implemented. RFagg uses the provided model to map all access links into the interfaces of a single virtual router. To add flows for a destination d into a datapath n with the best route to d , RFagg instructs each other datapath to forward traffic matching d over its link (or SDN-defined IP tunnel / LSP) to n . This was verified in the case study depicted in Fig. 1(b), where every AS 1000 edge switch is mapped to a single aggregation VM where a single BGP process terminates all AS’s peering sessions. This opposes the maintenance of four routers’ configurations in a legacy scenario, or the management of four virtual BGP processes, with their respective VMs, in previous RouteFlow deployments. Results show that forwarding through AS 1000 is correct.

6. Conclusions and work ahead

In this paper, we discussed IP routing under the visibility and control of SDN. Identified benefits relate to operational and traffic-engineering issues at both the control and data planes. The proposed RFCP allows for an incrementally deployable strategy to roll-out OpenFlow-enabled hardware following a controller-centric interworking approach.

Our efforts are devoted to continue the proof of concept implementation of new BGP-based services and the deployment of RFCP pilots with focus on the underserved mid-market (medium enterprises and regional ISPs). While this paper has mainly focused on BGP, ongoing investigations around OSPF and IS-IS promise good results to ease configuration management and protocol optimizations.

References

- Caesar, M., Casado, M., Koponen, T., Rexford, J., and Shenker, S. (2010). Dynamic route recomputation considered harmful. *SIGCOMM Comput. Commun. Rev.*, 40:66–71.
- Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A., and van der Merwe, J. (2004). The case for separating routing from routers. In *FDNA '04*.
- Filsfils, C. et al. (2011). Bgp prefix independent convergence (pic) technical report. Technical report, Cisco.
- Nascimento, M. R. et al. (2011). Virtual routers as a service: the routeflow approach leveraging software-defined networks. In *CFI'2011*, New York, NY, USA. ACM.
- Ramjee, R. and et al. (2006). Separating control software from routers. In *COMSWARE'06*.
- Van der Merwe, J. et al. (2006). Dynamic connectivity management with an intelligent route service control point. In *INM'2006*, New York, NY, USA. ACM.
- Wang, Y., Avramopoulos, I., and Rexford, J. (2009). Design for configurability: rethinking inter-domain routing policies from the ground up. *IEEE J.Sel. A. Commun.*, 27:336–348.