route selection, aBGP installs in n a prefix-length-based prioritized flow entry matching d with port-forwarding plus L2 re-writing actions. In every other DP, the installed flow actions are forwarding to DP n over their respective paths (e.g. IP/LSP tunnel). Hot potato can then be implemented by inspecting the RIB for multiple path entries and changing the IP prefix match action to inter-AS port-out interfaces. Experimental results show correct BGP operations and forwarding through AS 1000.

We have demoed RouteFlow with commercial OpenFlow devices from Pronto, NEC and IBM, and verified BGP and OSPF interoperability with Juniper MX and Cisco devices. A pilot instance of RouteFlow is used to control experimental traffic at Indiana University.[5] The experimental deployment is researching whether very different user interfaces can be built with SDN backends, and how to make campus network administration simpler to implement, more robust and consistent, and easier to manage by means of OpenFlow/SDN automation and abstractions.

6. DISCUSSION

We now discuss known challenges of the state of the art. Once resolved, which we expect shortly, the proposed architecture promises many new benefits.

6.1 Known Unknowns

Centralized BGP. Given the advances in (distributed) computational power and considering performance results of previous work [24, 26], RFCP-like systems shall be able to scale to maintain 10,000s of eBGP sessions, perform routing decisions for 100s of PEs, and process updates and store BGP routes from CEs. Further scalability can be achieved by implementing recent ideas [3] on RR BGP distribution based on chunks of address spaces rather than some fraction of speakers.

OpenFlow processing in datapath. While CPU power in the controller platform is abundant, cheap, and can be arbitrary scaled, CPUs of network devices limit the supported rate of OpenFlow operations. Publicly discussed numbers are in the order of few 100s of flow-mod/sec, and decrease with resource competing protocol tasks like flow-stat processing. We believe these numbers are due to unoptimized implementations. Nevertheless, they shall alert that large FIB updates could be very costly and thus the importance of PIC [8] implementations and pro-active backup flow installations, as discussed earlier in Section 4.

OpenFlow table size. Many commercial OpenFlow devices re-use existing TCAM enough for only 1000s of flow entries. Considering that we need at least one entry per destination subnet,[6] current HW configurations would be not enough for ISP production environments. Again, we believe this is a transient limitation and expect new OpenFlow optimized devices with larger flow capacity, potentially exposing existing L3 and L2 forwarding engines as tables in OpenFlow v1.X pipeline. On the positive side, the RFCP is in an excellent position to eliminate redundant state in the RIB and FIB by executing overlapping prefix suppression [18]. Alternatively, SW flow offloading decision engines could be used to save the precious flow space in HW [20].

High availability. Based on previously distributed implementations of RCPs [24, 26] and OpenFlow controllers [22], we are confident that the datastore-centric RFCP design has enough foundations to be fault-tolerant. Helpful techniques include RR-like route distribution among RFCP instances or maintaining multiple eBGP sessions with each CE. A failure of the BGP controller application would result in BGP session drops to CEs. To avoid service disruption, BGP route engines are not only physically distributed within each cluster, but moreover and completely transparent to the CE routers, we advocate the introduction of multiple RFCP clusters into the architecture by means of a newly defined BGP SHIM function in the OF edge switch. The BGP SHIM intercepts CE sessions and mirrors them (without need for parsing update messages) to multiple BGP RC clusters. That allows for very robust yet simple design which is arguably more robust than traditional PE local processing and RR advertisements.

6.2 The Promises

There is a number of advantages to the proposed IP split (hybrid) architecture, some applicable in general and inherent from OpenFlow/SDN, and some BGP-specific:

Simplified edge architecture. Relaxed requirements for every edge device to be fully capable of handling any new service or extension. Eliminated the need to process, store and maintain effectively the same set of control plane data and perform the same tasks across large number of edge platforms.

Lower cost and increased edge speed. Leveraging commodity switches and remote open-source routing software[7] decouples requirements for HW upgrades enforced by vendors due to end-of-life of particular OS. Increase in edge speed by closely following (with smaller CAPEX) the switching silicon latest technology curve, possibly optimized for flow switching.

Power of innovation leads to differentiation followed by new revenues. The ability to innovate in the network either by internal dev/ops teams or by vendor-independent third party products allows to differentiate the operator's services portfolio –without the need to convince vendors for support and practically sharing the innovations with other operators. Differentiation via uniquely customized network services allows much faster revenue opportunities for service providers.

BGP security, stability, monitoring, and policy management. As argued earlier, new ideas around BGP security become viable and cost-effective when executed in high CPU systems. Control plane stability (reduction of well-known BGP wave effect) can be increased by elimination of intra-domain BGP path oscillations [23]. BGP monitoring and reporting interfaces can be easily implemented since there is no need to collect all BGP "raw" feeds (aka original intention of BMP) from all border routers. APIs from the RFCP datastore enable view of entire BGP in an AS. Centralization of BGP policy management is a major gain in OPEX reduction and configuration error avoidance.

[5]Topology and an open-accessible UI are available here: http://routeflow.incntre.iu.edu/

[6]Conservative estimate assuming v.1.1 multiple tables that reduce cross-product state and enable efficient VRF.

[7]Customers may prefer solutions from major vendors instead of open-source alternatives. Such VM appliances allow for RIB events APIs that can feed the RFCP.

7. CONCLUSION AND WORK AHEAD

In this paper, we discussed how centralized BGP-speaking routing engines coupled with OpenFlow-based installation of IP-oriented flow rules leads to a new degree of control and enables a number of applications that are a real challenge (if not impossible) to realize in classic multi-vendor networks.

The proposed RFCP allows for an incrementally deployable strategy to roll-out OpenFlow-enabled devices following a "hybrid" controller-centric inter-networking approach. We reckon that a more "clean" and efficient replacement of legacy routing control protocols shall be pursued, leveraging the value of existing network configurations while addressing the potential impedance of admin incumbents.

We are devoting efforts on the proof of concept implementation of new BGP-based applications and the deployment of pilots for the under-served mid-market (SMEs and regional ISPs) for which we believe SDN technology may be very appealing. Already underway, enhancements to the platform include a converged dashboard GUI, coupled with historic data and a configuration repository available in the MongoDB. New RFCP services include load-balancing in multi-homed networks and feeding an ALTO server prototype. Ongoing investigations around OSPF and IS-IS promise advantages to ease configuration management and unlock protocol optimizations (e.g. IP Fast Reroute/LFA). Our roadmap includes LDP support to define MPLS paths, in addition to OAM considerations and OpenFlow v1.X advancements, such as IPv6 and overall extensibility.

8. ACKNOWLEDGMENTS

This work is partially supported by the GIGA Project (FINEP/FUNTTEL). Carlos N. A. Corrêa was supported by Unimed Federação RJ. The authors are thankful to all contributors of the RouteFlow project. For their valuable comments and suggestions, the authors are grateful to Maurício Magalhães, Jennifer Rexford, Brandon Heller, and the anonymous reviewers.

9. REFERENCES

[1] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In *NSDI'05*, 2005.

[2] M. Caesar, M. Casado, T. Koponen, J. Rexford, and S. Shenker. Dynamic route recomputation considered harmful. *SIGCOMM CCR*, 40:66–71, April 2010.

[3] R. Chen, A. Shaikh, J. Wang, and P. Francis. Address-based route reflection. In *CoNEXT '11*, 2011.

[4] J. Crowcroft. Reheating cold topics (in networking). *SIGCOMM CCR*, 39:48–49, March 2009.

[5] D. Saucez et al. Low-level design specification of the machine learning engine. FP7 ECODE Deliverable D2.3, Dec 2011.

[6] N. Duffield, K. Gopalan, M. R. Hines, A. Shaikh, and J. E. Van Der Merwe. Measurement informed route selection. In *PAM'07*, 2007.

[7] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *FDNA '04*, 2004.

[8] C. Filsfils and et al. BGP Prefix Independent Convergence (PIC) Technical Report. Technical report, Cisco, 2011.

[9] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Intelligent design enables architectural evolution. In *HotNets '11*, 2011.

[10] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: a strategy for transitioning to BGP security. In *SIGCOMM '11*, 2011.

[11] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4D approach to network control and management. *SIGCOMM CCR*, 35(5):41–54, 2005.

[12] E. Keller and J. Rexford. The 'Platform as a Service' model for networking. In *INM/WREN 10*, Apr. 2010.

[13] T. V. Lakshman and et al. The SoftRouter architecture. In *HotNets-III*, 2004.

[14] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch, and D. McPherson. Dissemination of Flow Specification Rules. RFC 5575, Aug. 2009.

[15] M. Motiwala, A. Dhamdhere, N. Feamster, and A. Lakhina. Towards a cost model for network traffic. *SIGCOMM Comput. Commun. Rev.*, 42(1):54–60.

[16] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. A. Corrêa, S. C. de Lucena, and M. F. Magalhães. Virtual Routers as a Service: the RouteFlow approach leveraging Software-Defined Networks. In *CFI '11*, 2011.

[17] R. Raszuk, C. Cassar, E. Aman, and B. Decraene. BGP Optimal Route Reflection (BGP-ORR). I-D draft-ietf-idr-bgp-optimal-route-reflection-01, September 2011.

[18] R. Raszuk, A. Lo, L. Zhang, and X. Xu. Simple Virtual Aggregation (S-VA). I-D draft-ietf-grow-simple-va-04.txt, September 2011.

[19] J. Rexford and J. Feigenbaum. Incrementally-Deployable Security for Interdomain Routing. In *CATCH '09*, 2009.

[20] N. Sarrar, S. Uhlig, A. Feldmann, R. Sherwood, and X. Huang. Leveraging Zipf's law for traffic offloading. *SIGCOMM CCR*, 42(1):16–22.

[21] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford. Network architecture for joint failure recovery and traffic engineering. In *SIGMETRICS '11*, 2011.

[22] T. Koponen and et al. Onix: A Distributed Control Platform for Large-scale Production Networks. In *OSDI '10*, Oct 2010.

[23] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. *SIGMETRICS '04/Performance '04*, June 2004.

[24] J. Van der Merwe and et al. Dynamic connectivity management with an intelligent route service control point. In *INM '06*, 2006.

[25] I. Varlashkin and R. Raszuk. Carrying next-hop cost information in BGP. I-D draft-ietf-idr-bgp-nh-cost-00, January 2012.

[26] Y. Wang, I. Avramopoulos, and J. Rexford. Design for configurability: rethinking interdomain routing policies from the ground up. *IEEE J.Sel. A. Commun.*, 27:336–348, April 2009.