An Industrial-Scale Software Defined Internet Exchange Point

Software-Defined Networking (SDN) has reshaped the design of many networks and is poised to enable new capabilities in interdomain traffic delivery. A natural place for this evolution to occur is at Internet Exchange Points (IXPs), which are becoming increasingly prevalent, particularly in emerging regions. Because many autonomous systems (ASes) interconnect at IXPs, introducing flexible control at these locations makes it easier for them to control how traffic is exchanged in a direct, and more fine-grained way. In previous work [2], we offered an initial design of a Software-Defined Internet Exchange Point (SDX) and showed how introducing SDN functionality at even a single IXP can catalyze new traffic-management capabilities.

Since we introduced SDX, many organizations, and networks have built or have begun to instantiate versions of this concept [1, 3]. Many of these deployments remain relatively small-scale or limited in scope. In this paper, we focus on design and implementation of iSDX, an industry scale SDX controller, that addresses two scalability challenges—fundamental to any SDX design.

The first challenge relates to how the control plane combines the policies of individual networks into forwarding entries in the data plane. Compiling traffic control policies expressed in a higher-level policy language to forwarding table entries can be slow since this process involves composing the policies of multiple participants into a single coherent set of forwarding-table entries. The fact that any change to BGP routing may change forwarding behavior further exacerbates this already slow process. Existing SDX designs trigger recompilation every time a BGP best route changes, which is not tractable in practice.

To scale the control plane, we introduce a new design that exploits the fact that each participant expresses its policy independently of other members, which implies that each policy can also be compiled and compressed separately, as well. This change enables more aggressive compression of the forwarding tables than is possible when a single controller compresses all of the policies and also allows compilation of participant policies in parallel. As a result, iSDX compiles the forwarding tables two orders of magnitude faster than existing approaches; the tables are also two orders of magnitude smaller, making them suitable for practical hardware-switch deployments.

The second challenge relates to the data plane: the number of forwarding table entries that might go into the forwarding table at an IXP switch can quickly grow unacceptably large. Part of the challenge results from the fact that the policies that each network writes have to be consistent with the BGP routes that other participants advertise, to ensure that an SDN policy cannot cause the switch to forward traffic on a path—never advertised in BGP. This process significantly inflates the number of forwarding table entries in the switch and is a considerable deployment hurdle. Large industrial-scale IXPs can have as many as 700 participants exchanging traffic for hundreds of thousands of prefixes; combined with the fact that each of these participants may now introduce policies for particular traffic flows, the number of forwarding table entries quickly becomes intractable. Although our initial design reduced the size of the forwarding tables, we show that the size of these tables remained prohibitively large for industrial-scale deployments.

To address the data-plane challenge, we introduce an efficient encoding mechanism where the IXP fabric forwards the packet based on an opaque tag that resides in the packets destination MAC field. This tag explicitly encodes both the next-hop for the packet and the set of ASes that advertise BGP routes for the packets destination, thus making it possible to remove this information from the switch tables entirely. This separation prevents BGP routing updates from triggering recomputation and recompilation of the forwarding table entries. Using features in OpenFlow 1.3 that support matching on fields with arbitrary bitmasks, we significantly reduce the size of this table by grouping tags with common bitmasks.

References

[1] J. Bailey, D. Pemberton, A. Linton, C. Pelsser, and R. Bush. Enforcing rpki-based routing policy on the data plane at an internet exchange. HotSDN, New York, NY, USA, 2014. ACM. (Cited on page 1.)

[2] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, pages 579–580, Chicago, IL, 2014. ACM. (Cited on page 1.)

[3] LightReading. Pica8 Powers French TOUIX SDN-Driven Internet Exchange, June 2015. `http://ubm.io/1Vc0SLE`. (Cited on page 1.)