Kumori: Steering Cloud Traffic at IXPs to Improve Resiliency

Antoine Fressancourt*†, Cristel Pelsser‡ and Maurice Gagnaire†
*Worldline R&D
†Computer Science and Network Dpt., Telecom ParisTech, Institut Mines-Telecom
23 avenue d'Italie, 75013, Paris - France.
Email: {antoine.fressancourt, gagnaire.maurice}@telecom-paristech.fr
‡Internet Initiative Japan
Email: cristel@iij.ad.jp

***Abstract*—After a few years of infancy, Cloud services have now gained enough maturity to be used to deliver an increasing number of critical services. To ensure the capacity of those services to survive failure events, major Cloud Services Providers (CSPs) deploy their platform in distant datacenters. The framework used to interconnect those datacenters is most of the time over-provisioned and costly to manage. In this paper, we present "Kumori", a SDN-based overlay architecture designed to give CSPs back control on their inter-datacenter connectivity. Using the iPlane dataset, we compare our architecture with the Resilient Overlay Network (RON), considered as a seminal project on Internet resiliency for the last ten years. Our results show that, depending on the CSP's size and connectivity strategy, our architecture either gives significantly shorter paths than RON in terms of latency or provides a similar service using a smaller overlay in terms of number of overlay nodes.**

***Keywords*—Software-Defined Networks (SDN), Overlay, Resiliency, Performance evaluation**

I. Introduction

Since Amazon launched the Elastic Compute Cloud in 2006, Cloud Computing has evolved significantly. While originally designed for non-critical services, Cloud infrastructures are now used for critical services ranging from banking to industrial production monitoring. Today, resilience is thus a preeminent requirement.

In order to prevent their services from going down if a whole datacenter (DC) fails or if a major disaster affects a whole region, Cloud Services Providers (CSP) usually deploy their services in several DCs spread around the globe. The services running in those distant DCs are synchronized and backed up using high capacity network links. Major CSPs such as Amazon or Google build their own network out of optical fiber links they deploy or buy from infrastructure operator to interconnect their DCs [1]. This strategy is not accessible to most CSPs because of its cost. As a replacement, smaller players rent dedicated private links from large Internet Services Providers (ISP). Thus, they often create a nearly-full mesh between the DCs they own composed of over-provisioned links they rent from several providers. The combination of over-provisioning and of the multi-vendor strategy is used to protect the CSPs from failures of network equipments (links and nodes) between two DCs.

In a previous paper [2], we have presented the design of an architecture to replace the traditional dual private link connectivity mesh used by CSPs. Our architecture is composed of elements located within and outside the DCs . It borrows concepts from Software-Defined Networking (SDN) to allow a CSP to control the path taken from one DC to another across the Internet. Between the DCs, the private links either built by the CSP or rented from ISPs are replaced by at least two best effort connections to the Internet provided by different providers and by an overlay network consisting in a set of nodes located at the Internet exchange points (IXPs). This overlay is controlled by a central entity monitoring the overlay and pushing dynamic routing instructions to the nodes in order to reroute traffic according to a given policy.

While we provided a detailed description of the overall architecture in this previous work, we kept major questions open: How does our architecture perform? How does it compare to the RON [3] architecture which aims at increasing the resiliency of internet connectivity between edge points using a decentralized architecture, focused on the edge? How many nodes in the overlay are needed in both architectures to reach a similar level of performance? Does our architecture provide similar benefits for every CSP or do some CSPs see more benefit from it?

In this paper, we first present the method we use to answer those questions. Our evaluation of the potential benefits of our architecture for inter-DC communications relies on data retrieved from the iPlane dataset [4] that summarize traceroute measurements done on the 15^{th} of February 2015. Using this dataset, we compare two overlay strategies: an overlay composed of edge nodes already belonging to the CSP, this is typical of RON or Detour [5], and an overlay consisting of nodes placed at IXPs representing our SDN-controlled overlay.

In our evaluation, we compared RON and the Kumori architecture using two performance metrics. First, for every pairs of Points of Presence (PoPs) belonging to a CSP in our set, we compared the length of the shortest path accessible using our architecture to the length of the shortest path accessible using the RON overlay. Then, we compared the minimal number of nodes that are needed in Kumori and in RON to be able to access the shortest possible paths between all the CSP PoPs

978-1-4673-8496-4/16/$31.00 ©2016 IEEE

pairs. Our intent was to compare Kumori and RON with regards to their cost of setup and operation.

In the remaining of the paper, we provide in Section II an overview of projects aiming at enhancing network resiliency. In Section III, we present the design goals of our architecture. We introduce the adopted metrics to compare Kumori with alternative approaches. Then, we give an overview of our SDN-based architecture and detail its inter-DC overlay. In Section IV, we justify our evaluation methodology. In Section V we detail and analyze the results we obtained before concluding in Section VI.

II. Related work

Improving the resiliency of Internet connectivity by means of overlays has been widely investigated for the last ten years. In the Detour project [5], Savage *et al.* made the observation that, in 30 to 80% of the failure cases in the Internet, there is an alternate path that has better characteristics in term of bandwidth, packet losses or round-trip time. The authors suggest using an overlay in which nodes are connected via tunnels, indirection being used to take advantage of those alternative paths.

Later, Andersen *et al.* designed a Resilient Overlay Network (RON) in [3]. Similarly to Detour, RON has been built to improve the resiliency of end-to-end connections in the Internet. In this project, the overlay is composed of nodes that actively measure the characteristics of the links between them in the overlay. This active measurement enables to react to failures very quickly. The main issue with this strategy is that measures have to be taken for each node pair. In their paper, Andersen *et al.* evaluate that this n^2 strategy limits the scalability of their architecture to roughly 50 nodes. A few years later, the authors suggested enhancements addressing parts of this problem in [6].

Gummadi *et al.* [7] have also addressed the issue of network resiliency while trying to address RON's scalability issue. They show that one-hop source routing can be used to route traffic efficiently around most failures. A random transit node in an overlay is used to route the traffic around a detected link failure. Source routing is used to control the way traffic is routed through this node. Compared to RON, permanent link monitoring is not used, yet the system achieves similar performance to that of RON in terms of resiliency and can be used in larger overlay networks. Meanwhile, the strategy adopted in this work cannot avoid last hop link failures if the destination node is not multihomed. In our architecture, we will tackle this limitation by requiring datacenters to be connected using multiple ISPs.

The three projects we presented at this stage rely on an overlay composed of nodes that are located at the edges of the network. In those project, the topology of the network is not taken into account, unlike in the work presented by Han *et al.* in [8]. In their work, overlay nodes are chosen in the network taking into account the network's topology. This topology information is also used in the construction of alternative detoured data paths to try to redirect traffic through only one node. In our architecture, we place the overlay nodes at IXPs, which are very specific locations in the Internet.

The possibility to control a router present at an IXP has been presented by Gupta *et al.* in SDX [9]. In this article, the authors present a Software Defined Internet Exchange combining the traditional peering using the Border Gateway Protocol (BGP) with the use of a SDN controller to support elaborated peering use cases such as application-specific peering, inbound traffic engineering or traffic redirection through middleboxes. This last use case is particularly interesting because of its proximity with our work. Yet, SDX is designed to be used by network operators bringing connectivity at the IXP rather than CSPs using SDN to redirect traffic among peering links they don't own. In our work, we want the overlay to be owned and controlled by the CSP. Using those nodes, the CSP can control the path taken by specific network flows, arbitrating between network operators as in [10]. In this project, Zhu *et al.* show the economical possibility for an overlay service provider to provide a better connectivity service in terms of QoS. This new actor provides a better network QoS by positioning multihomed routers at IXPs and dynamically selecting the best operator between the routers.

III. Kumori: a SDN-based network architecture for cloud resiliency

A. Design goals and performance objectives

The primary goal of our SDN-based network architecture is to reduce the cost of inter-DC connectivity while achieving a similar resiliency level as a full mesh composed of dual private links between datacenters. We want our architecture to be more flexible than deploying private links or MPLS circuits between the DCs. We also want to avoid the need to over-provision the links inter-connecting the DCs.

The first choice we made to reduce costs is to use the Internet rather than private links to exchange traffic between datacenters. Then, our architecture is used to enhance the resiliency of those Internet connections. Another way to reduce costs is to provide resiliency on a per flow basis: indeed, as shown for instance in [11], the flows between DCs are not of similar importance, and have variable requirements in terms of recovery time. Yet, in traditional methods using separate private links or MPLS circuits to ensure resiliency and reduce recovery time, all the DC traffic is considered, and the infrastructure is dimensioned in consequence. In our architecture, considering the flows differently will allow us to provide a faster recovery time to specific flows among the DC traffic. Thus, the architecture will be designed to provide fast rerouting for only a share of the total traffic.

The major drawback we want to overcome is the heavy reliance on Service Level Agreements (SLAs) to ensure a proper network resiliency. Those SLAs are often quite complex to enforce by the CSPs. SLAs only provide a compensation once a failure has occurred, while CSPs want means to react themselves. On the contrary, in our SDN-based architecture, the CSPs control the way they steer their connectivity and react to failure events.

978-1-4673-8496-4/16/$31.00 ©2016 IEEE

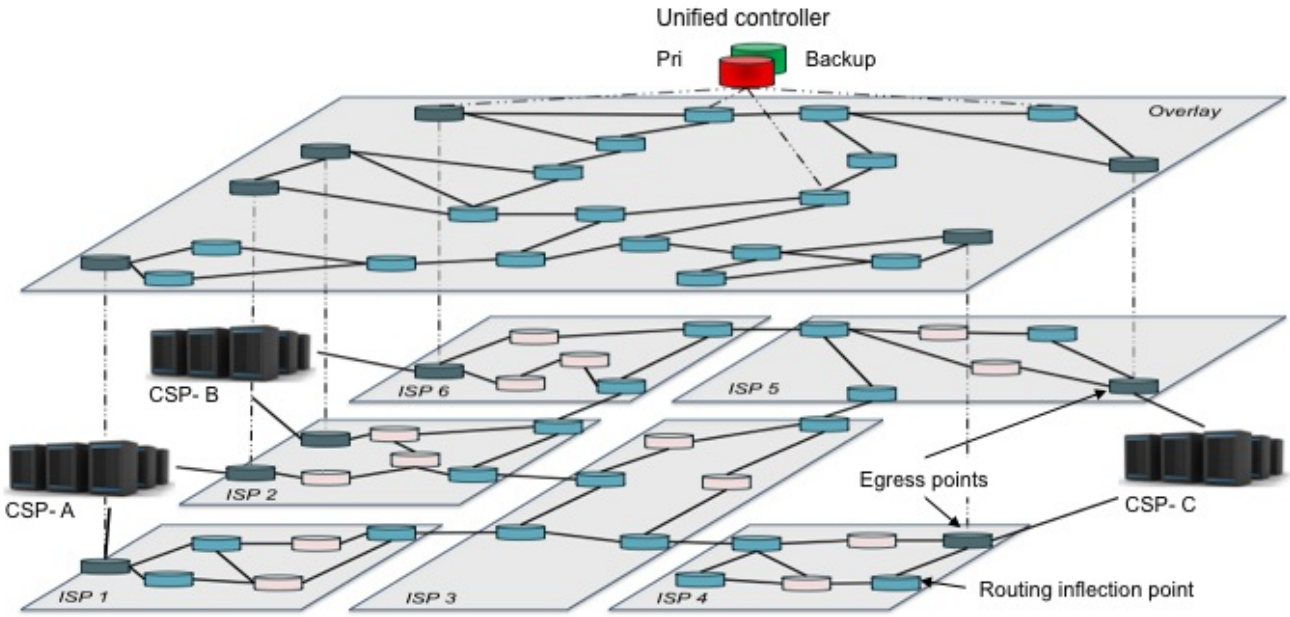

Fig. 1. Inter-datacenter architecture connecting three datacenters including the egress points, the routing inflection points and the unified controller

In its goals, our architecture shares the same ambition as RON: enhancing end-to-end network resiliency by providing the possibility to route traffic around failures using alternative paths. From a performance perspective, we want to perform better than RON on two aspects. First, we want to tackle RON's scalability limitation. This can be achieved by reducing the number of nodes needed in the architecture. Besides, we want the paths accessible via our architecture to be shorter or at least equal to the paths accessible using RON. In our evaluation, we use the link latency as a measure for the distance. Our aim is to be able to provide applications running in the CSP's datacenters with paths of smaller or equal delay to existing proposals.

B. Architecture overview

The Kumori architecture, which we presented in [2], is twofold. The two parts constituting Kumori aim at controlling the path taken by network flows between two servers first inside the CSP's datacenters and second between the datacenters in the Internet. In this paper, we focus on the inter-DC part. The main components of the architecture are depicted in Figure 1.

In Kumori, we aim to enhance the resiliency of inter-DC communications across the Internet by redirecting network traffic around failures. To achieve this resiliency objective, our architecture needs to provide a method to use a set of disjoint paths between its constituting elements. To maximize the amount of alternative paths accessible through our architecture, we will take advantage of the richness of the connectivity at IXPs. Indeed, Ager *et al.* [12] show that the number of interconnections between Autonomous Systems (AS) that are present at an IXP is underestimated, and accounts for a larger share of the inter-domain traffic than previously considered.

In addition to providing high connectivity to their members, IXPs are locations where significant innovations takes place. For example, Gupta *et al.* [9] proposed SDX, an SDN controlled IXP where members can define coarse grained policies to enable new services. The traffic exchanged at an Internet exchange point is managed via a software controller in order for peering policies and traffic management to be dynamic and adapted to contextual constraints.

The SDX project as well as recent SDN deployments at TouIX [13] , a French IXP and at the Wellington exchange [14], highlight IXPs desire to accomodate more services through the use of SDN concepts. It further assesses the technical feasibility of the software control of nodes located at an IXP. In the design of our Kumori architecture we take advantage of IXPs high connectivity and openness to new solutions.

For its inter-DC part, the Kumori architecture consists in a set of overlay nodes which are controlled by a central software element. The Kumori overlay is composed of two types of nodes: the egress nodes and the routing inflection points. The egress nodes are the points where the connectivity of the datacenters to the Internet is managed. Each egress node is associated to an ISP. In our architecture, we suggest for resiliency purposes that CSPs use at least two ISPs to connect to the Internet. Indeed, as we have seen in [15] or [8], this multihoming strategy is a mean to ensure connectivity resiliency all the way to the last hop.

The routing inflection points have an essential role in our architecture. They are located at various IXPs. From this privileged location, they can divert network traffic from the route advertised by the routing protocols in the various ISPs' networks. The aim is to be able to recover from failures faster than BGP. To that end, the overlay nodes steer traffic away from failed resources.

The routing inflection points and the egress nodes are coordinated by a central server, the unified controller. They are managed by the CSP. The CSP is thus able to enforce routing

policies for inter-DC traffic. For instance, after the discovery of a failure on a given path, it can actively steer network flows carrying real-time transactions away from the failure while letting the network's routing protocols deal with the redirection of network flows carrying backup data transfers. To react to a failure, the controller pushes alternative routing instructions to the routing inflection points and/or to the egress points. Those alternative paths might cross several routing inflection points to divert traffic around failures while keeping the paths as short as possible. To that extend, it is necessary that the controller has a view both of the overlay, and that it controls its routing. The resiliency provided by our architecture depends on the path diversity. The more distinct paths there are between the nodes in our architecture, the higher the potential disjointness is.

The routing inflection points regularly test the connectivity to the other routing inflection points and to the egress points to detect failures. These tests combine observations of active traffic flows and active measurements using packet probes for unused paths. The measurement results of those tests are sent regularly to the unified controller. The controller compares those measures to previous results to detect a drift in the packet loss rate or observed round trip time. This drift is interpreted as the sign of a degradation of the path. Besides, the absence of traffic between two routing inflection points, and the impossibility to exchange traffic probes between those inflection points is interpreted as a failure signal. Upon such a failure detection, the routing inflection points inform the central controller. This information is then used by the controller to reroute traffic.

In many ways, the routing inflection points behave the same way as SDN switches. Those overlay nodes need to look at the headers of packets belonging to a flow to divert and to encapsulate those packets in order to send them to the proper node. Those actions can be done by SDN switches as they can match packets against a set of header fields and rewrite packets on the fly. Besides, the information that the overlay nodes have to communicate to the central controller can easily fit in OpenFlow [16] messages. The main difference between our architecture and a more classical SDN network is that our overlay nodes are not directly connected, and they are not controlled by the administrator of the network domain they belong to.

IV. Evaluation methodology

In our evaluation, we compare Kumori to other network overlays aiming to reinforce the resiliency of connections between edge nodes in the Internet. The main difference between Kumori and those projects is the location of the overlay nodes: in Kumori, the routing inflection points are located at various IXPs while in other overlays the nodes are located at the edge. For the rest of the evaluation, the RON overlay is used as the edge overlay project to which Kumori is compared.

In this comparison, we would like first to determine whether Kumori and RON can provide alternative paths with similar performance characteristics. To that extend, we consider the delay of the paths. We compare the latency of the shortest paths accessible using both architectures. Besides, we want to evaluate the cost of deploying and operating Kumori compared to RON. In that regard, we make the assumption that this cost depends on the number of nodes participating in the overlay. We compare the number of nodes needed in Kumori and in RON to access the same number of alternative paths.

In our evaluation, we have chosen to use data extracted from the iPlane dataset [4] on the 15^{th} of February 2015. This dataset has two advantages. First of all, unlike simulations, it represents actual links that were observed and measured in the Internet on that day. Even if the dataset is not representing the whole Internet, it is rather significant. Besides, unlike data extracted from route servers, iPlane can reveal transit links that are used to route actual traffic while they remain invisible in BGP tables. Those two advantages make iPlane an interesting dataset to use. Yet, some work is needed to make it directly suitable to our study.

A. Building a graph from the iPlane dataset

The raw iPlane dataset takes the form of archives of *traceroute* measurements performed daily as well as some summarized datasets. Those summarized datasets gather all the inter-PoP links observed on a given day in the *traceroutes* and associate them an average of the performance metrics that have been measured. One of those summarized datasets gives the latency of the links between the points of presence (PoPs), the loss rates on those links, the association between observed IP addresses and PoPs and the Autonomous Systems (AS) the PoPs belong to. Those observations are performed every day since the inception of the iPlane project. In our evaluation, we use data summarizing the measurements done on the 15^{th} of February 2015. Using this data, we have built a graph representing the links between the PoPs that could be observed on that day. We have used the Python programming language to parse the iPlane dataset files and the igraph library to build the graph and manipulate it. The graph we obtained takes the form of an undirected weighted graph with 190,028 vertices representing the PoPs and 916,390 edges representing the observed inter-PoP links. We have chosen to use inter-PoP link latency as the weight associated to each edge.

In the iPlane dataset, the evaluation of the latency between PoPs raises two issues with regards to our evaluation. First, if traffic has been observed on a link but latency can't be evaluated accurately, a negative cost of -9,999 is given to the link. Yet, the igraph library doesn't accept negative values as an appropriate value to measure an edge cost. To solve this problem, we gave every node with a negative latency a weight equal to twice the maximum weight observed in the dataset. Second, some links have a latency equal to 0. Indeed, in iPlane, measures are rounded to the millisecond, and very fast links are given a zero cost. Yet, this cost doesn't take into account the switching cost at the PoP. To take this switching cost into account, we give every link with a zero cost a minimal cost equal to twice the latency measured by doing a *ping* on the

loopback interface of a linux server running Ubuntu 14.04 LTS, *i.e.* 0.5 ms.

B. Spotting IXPs and CSPs in the iPlane graph

Once our undirected weighted graph obtained, we need to spot the PoPs belonging to the CSPs or to the ISPs in order to determine and evaluate the shortest paths between the spotted nodes.

In order to identify the nodes belonging to the CSPs, we have selected the major global providers from market data gathered by Gartner, to which we have added some interesting regional actors. The result of this first identification step is a list of 13 companies. Then, we looked after the ASes managed by those companies in Hurricane Electric's BGP toolkit [17]. We obtained a list of 133 interesting ASes. At last, we looked in iPlane's dataset after the PoPs belonging to those ASes.

The identification of the nodes belonging to the various IXPs was more complex. Indeed, there is no centralized database of the existing IXPs, and by extension, no data about IP prefixes or ASes belonging to IXPs. Nevertheless, scarce data can be obtained from the PeeringDB [18], a database where network managers provide voluntarily information about their peering policy, or from Packet Clearing House [19], a non-for-profit research institute that operates routing measurement facilities at several IXPs around the world. We first cleaned the data we found in both databases and associated them in order to find the IP address subsets used by the various IXPs. Then, we found the PoPs that were present at an IXP by using the IP to PoP mapping given by the iPlane dataset.

As a result of this identification phase, we identified 1,604 PoPs belonging to a CSP and 2,177 PoPs present at an IXP out of the 190,028 vertices in the graph. In the next phase we keep all the vertices in the graph, and we use the identified PoPs to evaluate both RON and our Kumori architecture.

C. Graph study and measurements

To properly evaluate both our architecture and RON's capacity to route traffic between PoPs belonging to CSPs, we removed all the edges linking two nodes associated to the same CSP from the graph we obtained by parsing the iPlane inter-PoP latency dataset. By removing those edges, we make sure we compare our architecture to RON rather than to the CSP's interconnection strategy. Then, we looked after the shortest paths between all the CSP PoPs pairs, between the IXP PoPs pairs and between the CSP PoPs and the IXP PoPs. With the resulting shortest paths sets, we compared the paths obtained using the RON architecture and using our SDN-based inter-DC overlay using routing inflection points located at the various IXPs.

V. Results and Analysis

We compared our Kumori architecture with RON in two steps: First, we considered a large, imaginary CSP federating the CSPs PoPS we identified in our evaluation set. Then, we analyzed the result for each specific CSP in our evaluation set.

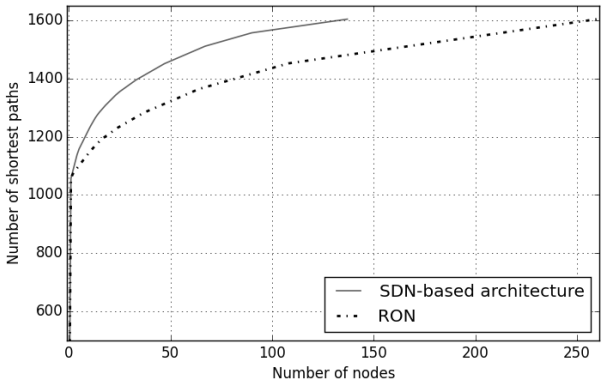

Fig. 2. Cumulative distribution of the number of nodes needed to access the maximum number of shortest paths

A. General results

In this section, we will first look at the results we obtained considering the entire set of PoPs belonging to a CSP whatever the provider they belong to.

First, our measurements show that the paths accessible using the Kumori architecture have a smaller or equal cost in terms of latency than the paths provided by a RON overlay for 1,255,950 CSP PoP pairs over the 1,285,606 possibilities. That represents 97.5% of the cases. If we consider a strict performance improvement with regards to latency, our architecture has better performance for 73,511 CSP nodes pairs over the 1,285,606 possibilities. It thus represents a strict performance improvement in 3.1% of the cases. This result shows that our architecture has similar performance as RON in the vast majority of the case, but doesn't show a drastic performance improvement most of the time.

After this first evaluation, we compared the number of nodes needed in each overlay network architecture to route data traffic between the CSP node pairs. Figure 2 presents a plot

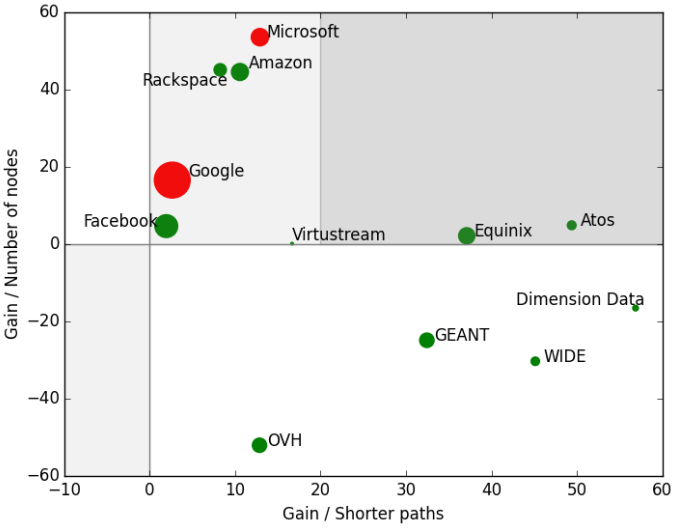

Fig. 3. Gain of Kumori Vs. RON in terms of path lengths and number of nodes needed to access the maximum number of shortest paths.

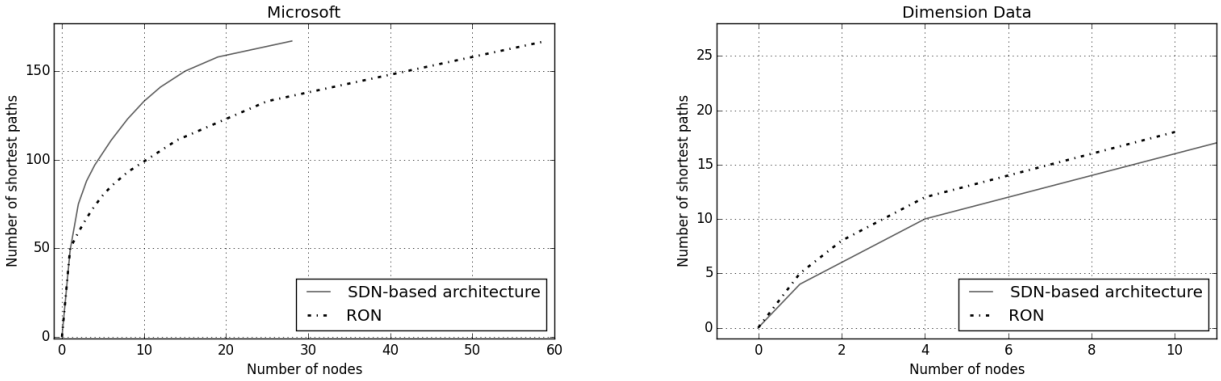

Fig. 4. Cumulative distribution of the number of nodes needed to access the maximum number of shortest paths for a large CSP (Microsoft) and a smaller CSP (Dimension Data)

of the cumulative distribution of the number of nodes needed in our architecture and in RON. The plot shows that 47% less nodes are needed to use all the shortest paths in our architecture compared to RON. If we consider 80% of the shortest paths between the CSP node pairs, only 15 nodes are needed in Kumori while 38 nodes are needed in RON. As the cost of operation of an overlay network depends on the number of nodes to deploy, this result shows that our SDN-based architecture can be less expensive to deploy and operate than a RON overlay.

B. Results analysis for specific CSPs

In the results we obtained in Section V-A, we considered that the PoPs belonging to the twelve large CSPs we selected belonged to the same, large CSP. Yet, those CSPs we have included in our study don't form an homogeneous group of actors, as the largest CSP in terms of PoPs in our set has roughly 200 times more PoPs than the smallest CSP. In this section, we study the gains provided by our architecture for each CSP in our set. We have also included two Cloud research infrastructures for the sake of comparison: WIDE and Géant.

In this study, we have compared our SDN-based overlay with a RON overlay for two metrics, and we plotted the results in Figure 3. On this figure, each dot represents a specific CSP. The diameter of the dot is proportional to the number of PoPs associated to the CSP. We have chosen to color the dots in red if the RON architecture can't be used for scalability reasons *i.e.* when the number of nodes needed to reach all the shortest paths is bigger than 50.

First, to compare the performance of the alternative paths accessible using both architectures, we evaluated the proportion of the paths accessible via our architecture that are strictly shorter than the paths accessible via RON. This measurement is the x coordinate of the plots representing the various CSPs on Fig. 3. Then, to compare the cost of operation of both architectures, we compared the number of nodes needed in Kumori to access all the shortest paths with the number of nodes needed in RON to access those shortest paths. We compute the difference between the two numbers, and divide this difference by the number of nodes needed to access all the shortest paths in RON. The resulting proportion is the y coordinate of the plots representing the various CSPs on Fig. 3.

On the figure, we can see two groups of points: one at the right of the figure, and another in the middle at the top of the figure. The first group of points is corresponding to the smallest CSPs in our evaluation set as well as the two Cloud research initiatives. For those CSPs, our SDN-based overlay architecture provides an access to more short paths than RON while using up to 20% more nodes in the overlay. On the contrary, the other group is corresponding to larger CSPs. Compared to RON, our architecture provides access to short paths using a lower number of nodes.

We explain those results by the difference between small and large CSPs regarding their connectivity. Large CSPs have optimized their network architecture to lower the cost to deliver network traffic to their customers. Amazon for instance proposes his customers to connect directly to his network at several points of presence through its Direct Connect offering. Our results show that the strategy adopted by those large CSPs is translated in a relative proximity of the CSP PoPs with the various IXPs. On the contrary, smaller CSPs often rely on their network connectivity provider to reach the Internet and their customers. Therefore, their PoPs are relatively farther from the IXPs than PoPs belonging to the large CSPs.

In this comparison between Kumori and RON, we can see that in every case, our architecture provides benefit on at least one dimension. The benefits that our architecture provides are quite different depending on the connectivity strategy adopted by the CSP. This result reinforces our wish to implement and deploy our overlay to evaluate it on the field.

VI. Conclusion

In this paper, we have presented Kumori, a SDN-based overlay to enhance the resiliency of inter-DC communication. In this architecture, resiliency is obtained by using the overlay nodes to reroute traffic more quickly than existing routing protocols. To evaluate the capacities of this architecture, we have built a graph representing the inter-PoP links in the Internet as revealed by iPlane, and compared Kumori with the RON overlay.

978-1-4673-8496-4/16/$31.00 ©2016 IEEE

Our numerical results show that the benefits the CSPs can obtain from Kumori depend on their size. For the smallest CSPs (*e.g.* Dimension data), the paths accessed via our architecture are shorter than those accessed using RON in at least 32% of the cases. For the largest CSPs (for instance Amazon), our architecture gives access to a similar set of shortest paths between the PoPs using up to 53% less overlay nodes. Thus, Kumori gives a solution to a major scalability issue related to RON. We explain those results by the difference between the connectivity strategies adopted by the CSPs. Large CSPs are well interconnected at IXPs to optimize their network costs while smaller CSPs still depend on large ISPs to connect their DCs to the Internet.

In the near future, we shall go deeper in the evaluation of the Kumori architecture. First, we want to revisit the results we obtained on path length using the packet loss rate of the inter-PoP links as a metric. Besides, our incoming studies will consist in implementing the Kumori architecture on a real testbed. More specifically, we want to measure the capacity of our central controller to detect network failures using measurements performed between the overlay nodes.

REFERENCES

[1] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug. 2013.

[2] A. Fressancourt and M. Gagnaire, "A sdn-based network architecture for cloud resiliency," in *Consumer Communications and Networking Conference (CCNC), 2015 IEEE 12th*, Jan 2015.

[3] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 131–145, Oct. 2001.

[4] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An information plane for distributed services," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2006, pp. 367–380.

[5] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, "Detour: Informed internet routing and transport," *IEEE Micro*, vol. 19, no. 1, pp. 50–59, Jan. 1999.

[6] D. Sontag, Y. Zhang, A. Phanishayee, D. G. Andersen, and D. Karger, "Scaling all-pairs overlay routing," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, 2009, pp. 145–156.

[7] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, ser. OSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 13–13.

[8] J. Han, D. Watson, and F. Jahanian, "Enhancing end-to-end availability and performance via topology-aware overlay networks," *Comput. Netw.*, vol. 52, no. 16, pp. 3029–3046, Nov. 2008.

[9] 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 *Proceedings of the 2014 ACM Conference on SIGCOMM*, 2014, pp. 551–562.

[10] Y. Zhu, C. Dovrolis, and M. Ammar, "Combining multihoming with overlay routing (or, how to be a better isp without owning a network)," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 839–847.

[11] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via yahoo! datasets," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1620–1628.

[12] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a large european ixp," *SIGCOMM Comput. Commun. Rev.*, pp. 163–174, Aug. 2012.

[13] "Pica8 powers sdn-driven internet exchange," Jun. 2015. [Online]. Available: http://www.reuters.com/article/2015/06/29/ca-pica-idUSnBw295365a+100+BSW20150629

[14] J. P. Stringer, Q. Fu, C. Lorier *et al.*, "Cardigan: Deploying a distributed routing fabric," in *SIGCOMM'2013 (Poster session)*, August 2013.

[15] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 353–364.

[16] "Openflow switch specification 1.4.0," Oct. 2013.

[17] "Hurricane electric's bgp toolkit." [Online]. Available: http://bgp.he.net

[18] "Peeringdb," https://www.peeringdb.com/private/index.php, retrieved Feb. 15, 2015.

[19] "Packet clearing house's full exchange point dataset," https://prefix.pch.net/applications/ixpdir/, retrieved Feb. 15, 2015.