Hybrid Cloud Integration of Routing Control & Data Planes

Prasun Kanti Dey and Murat Yuksel *
Dept of Electrical and Computer Engineering
University of Central Florida
prasun@knights.ucf.edu, murat.yuksel@ucf.edu

ABSTRACT

The Internet's routing infrastructure has always faced challenges due to flexibility needs originating from policy-driven path rules and scalability needs of an ever-growing number of control and data traffic. Recent Software-Defined Networking (SDN) designs elegantly separated the control plane from data plane and offered flexibility in path rule making, but various scalability issues emerged. Exploring a spectrum of designs, we propose a hybrid SDN routing architecture where cloud systems will keep *most* of the control plane functions and local router will keep the *least* of it while for data plane it will be vice versa. We highlight a hybrid separation where data plane *partially* resides in remote cloud while discussing the necessary and sufficient conditions to avoid possible loops.

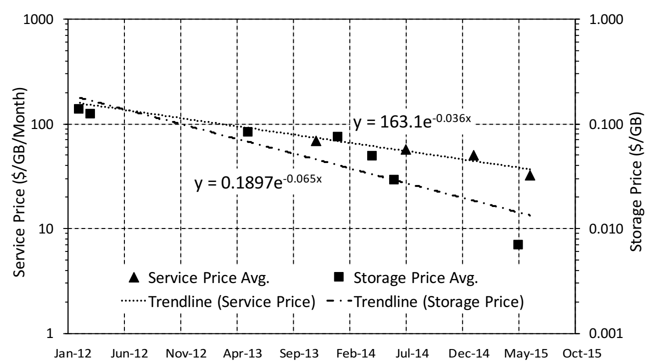

Figure 1: Cloud storage vs. service price trends

1. INTRODUCTION

For several decades, the cost of Internet's bandwidth halved yearly due to the advancement of optical systems like Dense Wavelength Division Multiplexing (DWDM). However, the cost of routing unit traffic is not keeping up with Moore's Law and decreased slowly compared to the increase of data traffic [17]. It is difficult to point to significant cost reduction in basic routing at the Internet core, because of the high expectation of packet processing capacities from a BGP router and the increasing line rates to accommodate the data traffic growth which is virtually doubling per year. It is necessary to add more switches and routers in the network backbone to attain parallelism which will not only improve the load balancing but also will act as backup route.

Software-Defined Networking (SDN) is poised to be an effective solution for overcoming this situation, which breaks the limitation of current infrastructure by separating control plane and data plane. With the development of open interfaces like OpenFlow [15], the trend to offload control plane complexity from core network components like router to software platforms for centralized optimizations [5] or outsource the control plane tasks to remote platform [14], such as cloud, are gaining their popularity.

In a service-oriented and software-driven market, ISPs are more interested in offering cheaper but better services using programmable network infrastructure that ensures quality of service (QoS) too. SDN's complete separation between the data and control planes model offers a great flexibility managing the overall network. As an example, OpenFlow v1.3 gives network administrator the luxury to configure QoS for the entire network from single controller and delegate the responsibility to update all the remaining routers to the controller rather updating each of them manually by himself, where OpenFlow v1.0 lacks this support [8]. However, credible performance and scalability concerns have already been raised as this may not be sufficient enough for supporting data-extensive applications such as multimedia streaming.

*This work was supported in part by NSF award CNS-1321069

CAN'16, December 12 2016, Irvine, CA, USA
© 2016 ACM. ISBN 978-1-4503-4673-3/16/12...$15.00
DOI: http://dx.doi.org/10.1145/3010079.3010085

On the other hand, since its emergence, cloud computing has been exhibiting a continuous decline in its pricing with increased competition which our finding also supports (See Figure 1). Another popular trend in cloud computing is its pay-as-you-go pricing. As people are using more cloud services and storage (in general), cloud providers are willing to provide more precise and detailed billing. These trends can make cloud platforms an ideal place to mix and match various computation and storage components to come up with a cost-effective hybrid designs involving cloud resources and network (hardware) components – potentially low level network functions like routing.

Considering the fact that complete separation between planes may not offer the best solution when it comes to scalability and flexibility [22], new architectures involving hybrid separation of control and data plane are already attracting the attention [18]. In this paper, we cover the critical architectural opportunities and challenges involved in such hybrid integration. In particular, we focus on the placement of data plane functions to a remote cloud platform and propose frameworks for failure-handling and loop-freeness in routing.

The remainder of the paper is organized as follows: Section 2 articulates the challenges of hybrid SDN model supported by cloud to gain more scalability and robustness. It also explores the privacy and security concerns while discussing the economics related to it as well. Section 3 covers the possible failure and loop scenarios for data plane delegation along with highlighting potential avoidance mechanisms. To conclude, Section 4 summarizes our work and discusses possible future work.

1.1 Motivation

In OpenFlow architecture, each flow has to consult initially with the controller for deciding their gateway port. This causes queuing delay in controller as the number of switches supported by a single controller goes higher, amount of forwarded flow towards that one increases linearly. Handling this growing queue becomes challenging for the controller. Earlier studies [10] states, NOX (an example of initial SDN controllers) can support 30K requests/sec optimally (which was not enough for Data Centers), while recent controllers are capable of supporting up to 10^5 requests/sec.

To detect link failure inside an OpenFlow architecture, either LOS (Loss Of Signal) or BFD (Bidirectional Forwarding Detection) is used. Both proactive (Protection) and reactive (Restoration) solutions has been proposed for failure recovery. Study [19] showed Restoration can not recover within 50ms time limit even though it requires less memory space in the flow table. On the other hand, Protection method asks for additional space to store backup paths while guarantees the service to be restored within 50ms. Van Adrichem et al. 2014 [21] introduced another fast (sub 50 ms) failover mechanism for OpenFlow controller that disproves the previous claim [19].

For reactive solution, a link failure recovery depends on when the switches request for a new flow update from controller. Proposed scheme for a faster restoration [19] assumes the failure in data plane. So, the controller keeps listening to any link failure and makes sure to compute another path to the affected end hosts.

On the contrary, Protection method uses GroupTable [9] concept, introduced in OpenFlow version 1.1, to maintain a redundant route. Each bucket of Fast-Failover group has watch port and/or watch group as parameter which monitors the up/down status of a port and chooses the next bucket if the liveliness of current bucket seems deemed down. Protection method uses Fast-Failover group type to avoid consulting with the control plane each time if a data plane link is failed.

Most of the earlier solutions focus to design a fault-tolerant architecture for small or medium sized network, while CORONET [12] describes a viable scheme that is scalable to bigger network. Their proposal also works on *data plane*, meaning if a link or switch goes down, the network can still regain its functionality.

2. HYBRID CLOUD INTEGRATION

With all the benefits that SDN offers, it is imperative to understand the scope of its flexibility and scalability. The basic premise of SDN technologies has been to offer high flexibility, i.e., programmability or how much the existing system can adapt to update the overall functionality without modifying much of the underlying infrastructure. Developing software-based abstractions, SDN has proven to be an extremely successful approach to managing many network components with low labor costs. Yet, its scalability is hindered by slower packet lookup times and increased overhead of abstractions affecting the agility of the network. Figure 2 illustrates this fundamental trade-off between scalability and flexibility of routing. The more we use specialized ASIC based (e.g., in high-end Cisco/Juniper/HP switches) designs, the more scalable routing becomes. But, such designs are more platform-dependent and restrict the modification of inner functionality. On the horizontal dimension, the routing design's programmability increases. That is, the network administrator can define finer granularity rules, the coarsest being 'per interface' and the finest being 'per packet'. Yet, there is a fundamental trade-off between these two dimensions of flexibility and scalability.

Most of the current SDN solutions are per-flow approaches (e.g., OpenFlow or Click) with a clean separation of control and data planes. This approach falls below the flexibility-scalability balance (the diagonal line in Figure 2) in the design landscape. We argue for a hybrid architecture, where local router (RX) will have *most* of the data plane functionality while offloading *partially* to a remote software proxy controller/entity (PRX) and vice-versa for the control plane. Hybridization of control plane, for example, means that calcula-

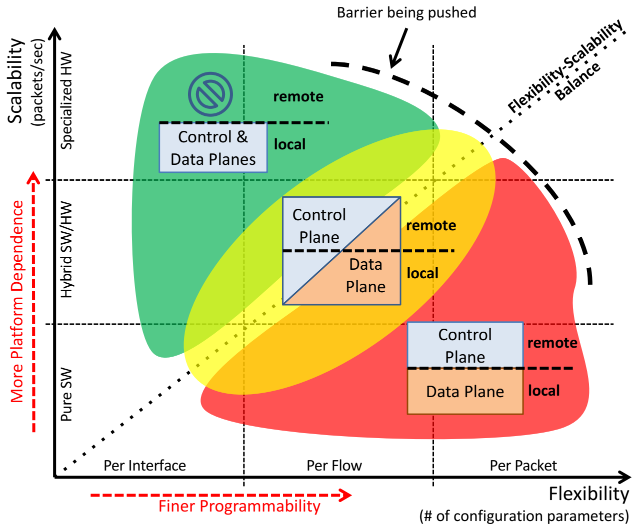

Figure 2: Routing scalability and flexibility trade-offs

tion of most of the shortest-paths will be done at PRX while a small portion of them (with high importance) at the RX. Such trend of getting a small portion of control plane tasks at local router RX is already taking place in recent versions of OpenFlow. Simple counters and statistics collection on flow table entries is now being done at local router RX and reported to controller [7], rather than relying on polls from the controller.

The hybridization in data plane means that some of the packet forwarding tasks will potentially be delegated to PRX (discussed in Sec 3). The frequency of such delegation will depend on the size of the partial Forwarding Information Base (FIB) at the local router RX as well as FIB caching techniques determining which entries from the full FIB table at PRX will be maintained at the partial FIB in RX. We argue that delegation of data plane tasks to the remote PRX will be a key step in advancing offloading of low-level networking functions to powerhouses like cloud computing platforms.

We think that such hybrid designs of the control and data plane tasks will be more attractive as cloud computing platforms are becoming more central to the network management. So, we consider the hybrid designs where PRX is located in a cloud platform which *may* be (i) reachable over the public Internet, (ii) operated by a separate business in return of "cloud-assisted routing (CAR)" [3] service fee, and (iii) tens of milliseconds away from RX. In the following sub-sections, we will discuss the key issues in such hybrid integration of cloud into routing functions.

2.1 Flexibility

Pushing the full control plane to separate third-party cloud providers offers new flexibility advantages in managing many routers. SDN provides better management of network middleboxes by eliminating the requirement of static policy setup. One prominent issue regarding middleboxes in SDN architecture is their proprietary behavior, which in return prevents the controller from setting proper forwarding rules due to the lack of overall topological information. However, as proposed [20], cloud can act as a trusted third-party for providing a secured solution for such problems.

As discussed in [18] and portrayed in Figure 2, specialized ASIC based routers offers tremendous scalability as they are made specifically to handle 10+ Gbps traffic. Because of the proprietary and black-box nature, they are not programmable, which is a huge drawback regarding their flexibility. Instead, typical SDN routers running on OpenFlow (or even Click) routers are very flexible and open for any sort of modification to support per interface or even per flow based packets; albeit, digging into each packet is highly computation-intensive job which may not be supported by any individual routers. With the power of parallelism in computation and being backed up with a huge memory and processing power, cloud platforms can host this type of control plane task very efficiently.

Elasticity in on-demand resource allocation makes cloud a natural choice for supporting seamless integration of multiple ASes with different programmable interfaces. Another obvious optimization that can be achieved from hybrid model if multiple ASes are being supported by the same cloud. Cloud provider itself can find inter-AS shortcuts and delegate the packets to destination. Unfortunately this kind of multi-hop optimization is restricted in legacy SDN due to its clean separation.

2.2 Performance and Scalability

Cloud computing, despite being mainly data center centric, intends to offer a distributed computing infrastructure that can be scaled globally. We follow the evident trend of cloud-based networking and explore the possibility of circumventing non-trivial complexity and cost of legacy routers at the Internet core by using cloud services. Current SDN designs face two major scalability problems:

(1) *Fitting ever-growing BGP routing entries into FIB tables* remains an age-old challenge [1]. Due to the temporal and spatial data locality [11], it would be interesting to observe the performance of a hybrid approach where a local RX keeps a *partial* of the entire FIB table storing the *full* FIB in a remote cloud. If engineered well, only a small portion of the data traffic will have to be delegated to PRX. If this delegated traffic is kept under control, the delay due to the delegation and its potential effects on fairness to data flows could be minimal.

(2) *Scaling controller for larger number of switches and lower latency.* Tapering of the overload on controller and the latency in communicating between controller and switches are critical challenges. Till now, OpenFlow involves the controller to make decision for every new flow that comes to the switches. As discussed in Section 1.1, existing OpenFlow controller (e.g., NOX

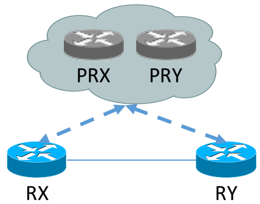

Figure 3: Single Cloud Vendor hosting multiple proxy

[6]) can support up to 100K requests/sec [10]. Cloud can be a perfect fit to scale up the amount of supported requests coming from a bigger network with more switches.

In terms of latency, our hybrid approach can potentially degrade the performance due to the delegated traffic flows having to travel extra (from RX to PRX) to reach the cloud. As new players join the business of providing new cloud-assistance to routing, it is more likely to identify a cloud that is closer and have higher quality. For instance, one can use CloudCmp [13] to identify the closest cloud provider with lowest possible latency in response time for various computation and make sure local router delegates the packets towards that cloud if necessary. On the positive side, such delegation of data traffic effectively makes routing a multi-path one and the overall end-to-end delay may even improve, as we have observed in our recent experimentation [3]. With careful engineering, we feel the drawbacks can be solved in return of more overall performance benefit.

2.3 Privacy, Security and Fairness

Since PRX needs to communicate control information with RX, it is mandatory to maintain a secure tunnel between them. A separate TCP session or a GRE tunnel can be implemented between them to achieve the security. Further, originating from SDN's trend of "centralized brain", PRXs may be treated like a honeypot by attackers due to their location's (i.e., the cloud) single point of control. Denial-of-service attacks will be more challenging to deter from these cloud points since they will likely not just serve routing assistance but other more generic services too. Dedicated sessions from PRXs to their corresponding RXs may be the only practical solution for these security challenges.

Another vulnerability may arise due to the basic SDN design which involves supporting multiple ASes from a same cloud vendor. Provider needs to isolate the service layer agreements (SLA) of different ASes and has to come up with a solution similar to FortNox [16] for preventing possible inconsistency. For a scenario presented in Figure 3, cloud provider can resolve inter-domain policy conflicts with the capability of monitoring and troubleshooting any issues more easily as multiple ASes share their policy with it.

Another relevant issue is the fairness to traffic flows. Since, some flows will be delegated to PRX, they will experience more delay. To comply with net neutrality expectations, operators configuring the hybrid cloud integration will need to ensure fair treatment of all flows irrespective of content, source of origination and size. RX should follow non-discriminative *Traffic shaping* to support common practice of treating the flows equally. As previously found [11], there exists high locality (10% popular prefixes are responsible of 97% of traffic), and the designer should try to support these flows locally. Yet, the fairness concerns require going beyond straightforward selection of the heavy hitter flows but also consider caching small hitter flows belonging to unpopular prefixes. The overall routing experience of the flows (heavy or small) should be roughly similar. Techniques like punching holes and super-netting will have to be actively devised to attain such balance.

2.4 Economics

In 2016 January, Microsoft made 17% price cut [2] for their Azure platform following the trend of Amazon and Google. We have collected the prices for 3 of the most popular cloud storage providers (Microsoft, Google, Amazon) and 11 of the service providers (IBM, GoGrid, RackSpace, AT&T as well) and plotted their average in Figure 1. We can observe the declining trend while there is not enough concrete evidence of router hardware price reduction which backs up our claim to offload control plane functionality to cloud.

Market leaders are building their high-computation capable infrastructure for supporting wide-range of applications. Features like load-balancing and auto-scaling are already becoming a trend by the major providers like Amazon and Microsoft. All these new features will substantially pave the way for current SDN architectures to further integrate cloud to host their control and data planes. Earlier work [4] explored the viability of cloud assistance in hybrid architecture by comparing cost models for traditional and hybrid routing.

As prices go down, cloud providers are investing into offering more value-added services with a very little tweaking in their infrastructure. This will encourage them to implement routing features on cloud systems, where fixed size virtual routers, PRX, can be purchased for a very small payment (quite similar to currently accessible cloud-based virtual machines).

3. DELEGATION OF DATA PLANE

Legacy SDN designs did involve delegation of control plane tasks to remote platforms, but hybrid integration of routing functions with cloud support imply delegation of data plane tasks. We envision delegation of data packets to PRX, which brings new challenges and opportunities.

3.1 Better Failure-Resiliency

Delegating data packets to cloud platforms and letting them further forward the packets open up interesting opportunities in terms of robustness of routing. As shown in Figure 4, hybrid cloud integration can mirror

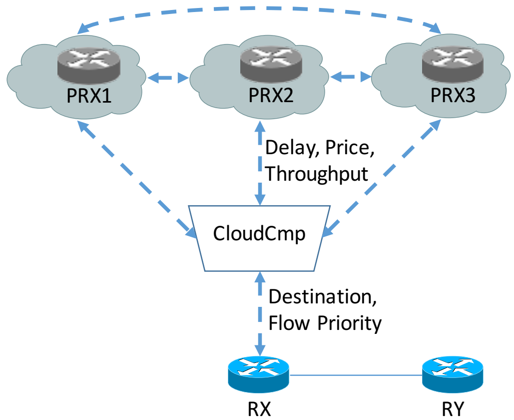

Figure 4: Hybrid Design using Multiple Cloud Vendors

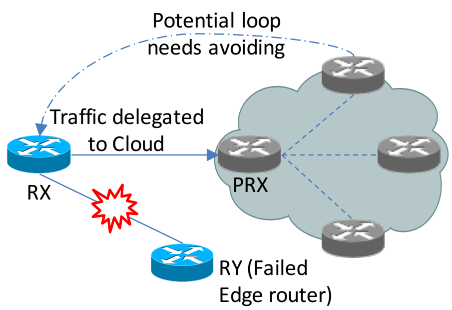

Figure 5: Traffic re-route due to failure

PRX in multiple clouds. Then, *RX* (or an intermediary similar to CloudCmp [13]) can choose the best mirror *PRX* based on destination or priority of the flow, and service prices of the cloud providers hosting the mirrors.

In conventional routing, if a router fails, significant amount of traffic needs to be rerouted to an alternate path or through nearby devices. There are already a lots of well-researched proposals to achieve better resilience by analyzing link failures, designing over-provisioning models for legacy network. Figure 5 shows the packet delegation procedure in hybrid architecture where *RX* delegates packets towards its proxy when the neighboring link with router *RY* (this can be another hybrid SDN router or a legacy router) fails.

As in Figure 3, a single cloud provider can host proxy routers for multiple routers. The packets delegated to *PRX* can be handed to *PRY* if they are destined to *RY*. Cloud providers can improve the overall end-to-end forwarding delay of such delegated packets if they serve more such hybrid routers – perhaps to the extent that delegated packets may arrive faster. Such centralized role of clouds in data plane tasks may significantly improve the overall robustness and efficiency.

Further, delegation of control plane tasks to cloud will surely offer an opportunity of both inter- and intra-domain routing optimizations as routers of multiple ASes may be supported by an individual cloud provider or multiple proxies from a single AS may be supported.

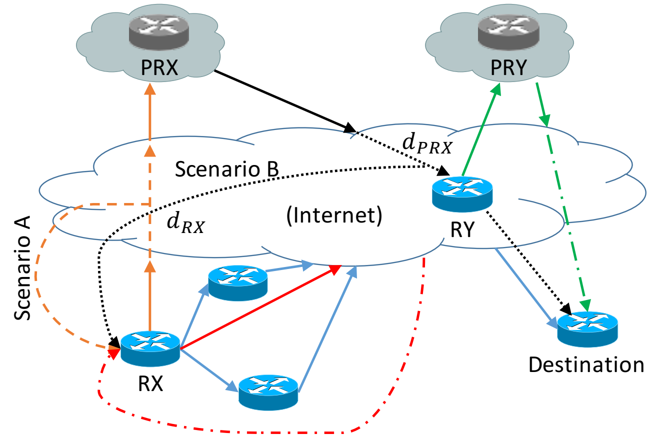

Figure 6: Possible Loop scenarios

3.2 Attaining Loop-Freeness

With the development of SDN designs with clean control plane separation, a prominent issue arises is whether there can be such situation when a packet never reaches to its final destination due to loops [7]. It is always adverse to have a network inconsistency that has been generated due to the complex chaining of instruction. Similar concerns apply for the proposed hybrid SDN architecture. While designing such scheme, network designers should be careful about possible inconstancies, particularly when the data traffic delegated to *PRX* is using public Internet routing.

Considering existence of multiple hybrid SDN routers on a path, two possible loop scenarios may arise due to data traffic delegation:

3.2.1 Scenario A: While Reaching PRX From RX

While delegating to *PRX*, a loop may be established if the Internet routing chooses RX as the next-hop. This scenario is not possible unless there is an inconsistency in the Internet routing. Assuming that the delegated traffic will go through a GRE tunnel, the shortest path towards the destination of that tunnel (i.e., *PRX*) will not traverse *RX* in steady-state routing.

3.2.2 Scenario B: Between PRX and Destination

Once delegated to *PRX*, *PRX* will forward the data traffic towards its destination via shortest-path Internet routing. As shown in Figure 6, such delegated traffic may get delegated again to another proxy router *PRY*. In general, such traffic may be looped if the distance of destination from *RX* (d_{RX}) is less then from *PRX* (d_{PRX}). Further, a loop exists if the shortest path from *PRX* to destination (SP_{PRX}) includes the shortest path from *RX* (SP_{RX}). So the following conditions should never be met to make sure a loop-free architecture:

1. *Necessary condition:* $d_{PRX} \geq d_{RX}$
2. *Sufficient condition:* SP_{PRX} includes SP_{RX}

One approach to prevent such loops is to *only* allow delegation of prefixes to PRX for which PRX's forwarding table has strictly less cost towards the destination, i.e., $d_{PRX} < d_{RX}$. This will assure that the necessary condition for a loop is never met. Yet, such preventive design may be too restrictive in terms of caching of prefixes in RX and limit traffic engineering possibilities.

4. SUMMARY

Reliability in communication links between controller and the hardware switches in existing SDN still remain unanswered. If a trusted third party like cloud can take over a significant portion of control plane functionality and provide a secure communication via public Internet service, concerns like switch-controller link failure, queuing delay in controller can be mitigated easily. Recent trends in cloud availability and decreasing service prices point us towards a more cloud-integrated networking, potentially spanning data plane tasks as well. We have proposed a hybrid cloud integration of routing functions in both control and data plane tasks. Rather than a clean separation of control and data planes, we argued for a hybrid SDN architecture aiming to balance the trade-off between flexibility and scalability. For such hybrid designs, we outlined the potential challenges and opportunities, particularly matters arising due to delegation of data plane tasks to a remote cloud platform.

5. REFERENCES

[1] BGP Routing Table Analysis Reports, 2016. Available online: http://bgp.potaroo.net.

[2] Cloud price cuts continue with azure, but don't expect a trend, January 2016. Available online: http://tinyurl.com/zgroxpl.

[3] P. K. Dey and M. Yuksel. CAR: Cloud-Assisted routing. In *2016 IEEE Conference on NFV-SDN*, Palo Alto, USA, Nov. 2016.

[4] P. K. Dey and M. Yuksel. On the breakeven point between Cloud-Assisted and legacy routing. In *IEEE CloudNet '16*, Pisa, Italy, Oct. 2016.

[5] N. Feamster, J. Rexford, and E. Zegura. The road to SDN: An intellectual history of programmable networks. *ACM Queue*, 2013.

[6] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM CCR*, 38(3):105–110, 2008.

[7] F. Hu, Q. Hao, and K. Bao. A survey on software-defined network and openflow: from concept to implementation. *IEEE Comm. Surveys & Tutorials*, 16(4):2181–2206, 2014.

[8] A. Ishimori, F. Farias, E. Cerqueira, and A. Abelém. Control of multiple packet schedulers for improving qos on openflow/sdn networking. In *2013 Second European Workshop on Software Defined Networks*, pages 81–86. IEEE, 2013.

[9] R. Izard. How to Work with Fast-Failover OpenFlow Groups, April 2016. Available online: http://tinyurl.com/nb7schn.

[10] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li. Software defined networking: State of the art and research challenges. *Computer Networks*, 72:74–98, 2014.

[11] C. Kim, M. Caesar, A. Gerber, and J. Rexford. Revisiting route caching: The world should be flat. In *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, pages 3–12, Seoul, Korea, 2009.

[12] H. Kim, M. Schlansker, J. R. Santos, J. Tourrilhes, Y. Turner, and N. Feamster. Coronet: Fault tolerance for software defined networks. In *2012 20th IEEE ICNP*, pages 1–2.

[13] A. Li, X. Yang, S. Kandula, and M. Zhang. CloudCmp: comparing public cloud providers. In *Proceedings of ACM Internet Measurement Conference*, November 2010.

[14] T. A. Limoncelli. OpenFlow: A radical new idea in networking. *ACM Queue*, June 2012.

[15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM CCR*, 38(2):69–74, 2008.

[16] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu. A security enforcement kernel for openflow networks. In *Proceedings of the first workshop on Hot topics in SDN*, pages 121–126. ACM, 2012.

[17] L. G. Roberts. A radical new router. *IEEE spectrum*, 7(46):34–39, 2009.

[18] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao. Are we ready for SDN? implementation challenges for software-defined networks. *IEEE Com Mag*, 51(7):36–43, 2013.

[19] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester. Openflow: Meeting carrier-grade recovery requirements. *Computer Communications*, 36(6):656–665, 2013.

[20] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of ACM SIGCOMM*, Helsinki, Finland, August 2012.

[21] N. L. Van Adrichem, B. J. Van Asten, and F. A. Kuipers. Fast recovery in software-defined networks. In *2014 Third European Workshop on SDN*, pages 61–66. IEEE, 2014.

[22] S. Vissicchio, L. Vanbever, and O. Bonaventure. Opportunities and research challenges of hybrid software defined networks. *ACM SIGCOMM CCR*, 44(2):70–75, 2014.