An Enhanced Path Computation for Wide Area Networks based on Software Defined Networking

Djamel Eddine Kouicem, Ilhem Fajjari and Nadjib Aitsaadi‡
Orange-Labs, F92320, Chatillon, France
djameleddine.kouicem@orange.com, ilhem.fajjari@orange.com
‡University Paris-Est, LIGM-CNRS UMR 8049, ESIEE Paris: F-93162, Noisy-le-Grand, France
nadjib.aitsaadi@esiee.fr

***Abstract*—Global IP traffic is forecast to triple by 2020 to reach 2.3 ZB per year. Such an explosion will inevitably be the catalyst of Operator infrastructure transformation. In this context, SDN is the technology that is shaping the future of carriers' networks. It offers the opportunity to implement more powerful control algorithms. In this perspective, we put forward a SD-WAN architecture to enhance the network resources allocation and hence improve the QoS of distributed applications. The main idea is to take profit from the accurate network view provided by the controller to optimize the flows routing in WAN environments. To do so, we formulate the path computation problem as an Integer Linear Program by taking into consideration both network application requirements and the network occupation status. The problem is then resolved in a polynomial time leveraging the branch-and-cut algorithm. Results obtained based an experimental platform show that our ONOS SDN framework outperforms the most prominent related work solutions in terms of network consumption and applications satisfaction level.**

Keywords: Routing, WAN, SDN, ONOS, Distributed Cloud

I. INTRODUCTION

The emergence of a new generation of services has encouraged operators to make every effort to grow and consolidate their infrastructures. The main goal is to offer richer services with a better quality of experience for more demanding consumers. The annual global IP traffic is expected to triple over the next four years to reach 2.3 ZB by the end of 2020[1]. Consequently, new innovative techniques are mandatory to face the exponential increase of volume and number of sessions. In this context, operators look for adding programmability and agility to their network infrastructures to guarantee ultra-low latency, high bandwidth and real-time access to their services.

It is undeniable that Cloud has become a pillar of operators infrastructure. Indeed, the emergence of new kinds of cloud-based technologies speed up the growth of their physical data centers while ensuring higher flexibility and better performances. Whereas, the multiplication of data centers raises new challenges especially for distributed cloud based services. For example, Distributed NFVs [1], Cloud-RAN [2], Edge computing based applications [3] require high connectivity between data centers to ensure the capability of transmitting high bandwidth of flows and live migrating of virtual machines. It is straightforward to see that the success of a distributed cloud-ready environment deployment depends on the capability of Wide Area Networks (WAN), carriers' core assets, to efficiently deliver workloads. Several approaches have emerged aiming to improve the network capacity based on Traffic Engineering techniques such as MPLS-TE [4], RSVP-TE [5], ECMP [6], etc. However, despite the achieved efficiency, the latter proposals are still complex to configure and struggle to scale. That is why fulfill carriers' requirements for faster services provisioning is failed.

To overcome the above weaknesses, an interesting approach would be to separate the forwarding and the control planes. This is the main idea behind the concept of Software Defined Networking (SDN) [7] [8] [9] which is a revolutionary technology shaping the future of carriers' networks. It considerably reduces the complexity of managing the network infrastructure while providing tremendous computational power compared to legacy devices. Actually, SDN controllers offer the opportunity to implement more powerful routing algorithms thanks to the real-time centralized control leveraging an accurate view of the network.

In this paper, we design a SD-WAN architecture ideal for supporting new generation of applications with vastly different requirements such as: bandwidth, loss and latency. Thousands of such applications are i) running across multiple data centers interconnected via dedicated private WANs and ii) are potentially maintaining thousands of individual active connections to distant servers. It is clear that handling such a huge traffic within a WAN need powerful path computation and optimization techniques able to enhance the quality of service.

Our SD-WAN architecture is built on ONOS controller [10]. It makes use of i) BGP-LS [11] to collect network resource information, ii) PCEP [12] to ensure the communication between the controller and network element, iii) Segment Routing protocol [13] to ensure the assignment of labels and iv) an optimized path computation algorithm. The main objective is to improve both the application satisfaction level in terms of bandwidth and the resource usage of the network infrastructure.

II. SDN-BASED WAN ARCHITECTURE

Our objective is to build an efficient SDN based architecture able to optimize the path computation in WAN environment to enhance the QoS of distributed applications. The overall framework of the proposed architecture is illustrated in Fig. 1. Consistent with SDN principles, our architecture

---

[1] Cisco VNI Forecast and Methodology, 2015-2020

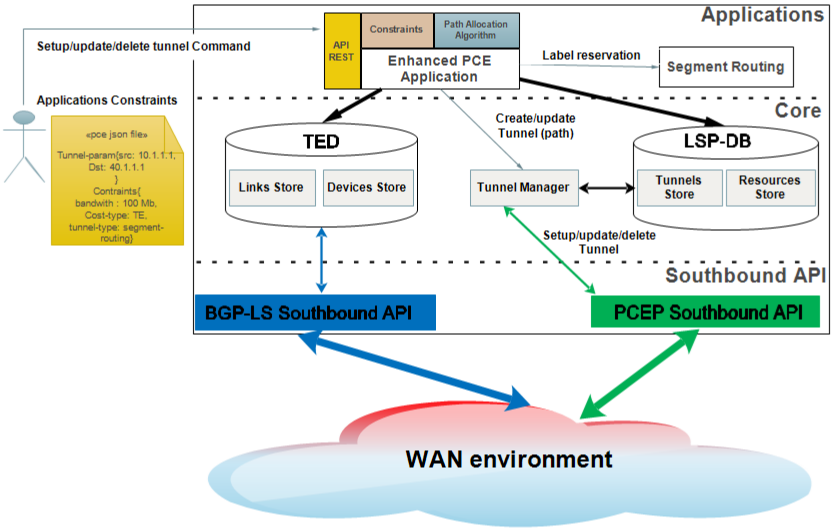

Fig. 1. SD-WAN global architecture

relies on three main layers: *i) application*, *ii) core* and *iii) southbound*. As depicted in Fig. 1, the southbound layer is based on two main protocols: BGP-LS and PCEP.

BGP-LS as described in Fig. 2 is responsible for the collection of link-state and traffic engineering information from the underlying WAN. Then, the collected metrics are shared with the SDN controller making use of BGP routing protocol. It is worth noting that the Interior Gateway Protocol (IGP) is responsible for the routing intra Autonomous System (AS) such as: OSPF, ISIS, etc. BGP-LS provides an abstraction layer and translates to its own model the IGP's native information. Such an abstraction is achieved thanks to the new BGP Network Layer Reachability Information (NLRI) encoding format [14]. Indeed, by creating new NLRI, the whole IGP information can be carried over BGP while leveraging BGP-speakers which are already participating in IGP.

In this paper, the IGP assumed is Open Shortest Path First Traffic Engineering (OSPF-TE) [15]. BGP-speaker collects the detailed network performance information. Then, the latter are distributed whether directly or through another BGP-speaker to the SDN controller using Type-Length-Value (TLV). Note that network metrics such as link propagation, latency, delay variation, loss, residual bandwidth, utilized bandwidth are retrieved from the Link State Data Base (LSDB) as defined in RFC 7471. It is straightforward to see that by considering these metrics in the path computation process, the decision will be more cost effective and scalable.

The SDN controller acts as stateful active Path Computation Element (PCE). The latter stores in the LSP-DB located in the core layer (see Fig. 1) the list of established LSPs and their corresponding parameters such the route and the reserved resources. We recall that also the collected information related to traffic engineering are stored in LSP-DB. By refreshing the synchronization of LSP-DB according to the WAN state, PCE is able to further optimize the path computation. The LSP-DB is populated and updated by PCEP [16]. In addition, the maintenance of LSP-DB is ensured by the controller-Path Computation Client synchronization requiring the exchange of LSP status updates (Path Computation Client). We recall that the PCC correponds to the client application requesting a path computation to be performed by a PCE. Thanks to the active mode of the controller, triggers can configured to add and/or remove LSPs with respect to the application request and network state. Through the SDN controller API, an administrator may request a path computation between source and destination nodes in one or several areas. Such a request will be handled by the PCE entity of the controller and then sent to the corresponding PCC. The synchronization between the controller and the PCC is achieved thanks to the exchange of PCEP Report message, including the LSP Explicit Route Object (ERO) and its attributes. The instantiation is realized by means of PCEP Initiation message (PCInitiate), which specifies all the attributes of the path to be instantiated (ERO, bandwidth, latency, etc.). The PCInitiate is used to trigger the removal of an existing LSP.

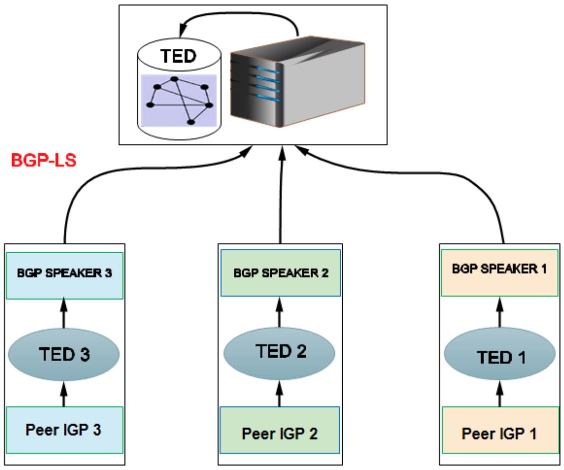

Fig. 2. BGP-LS functioning

The core layer is composed of three main components:

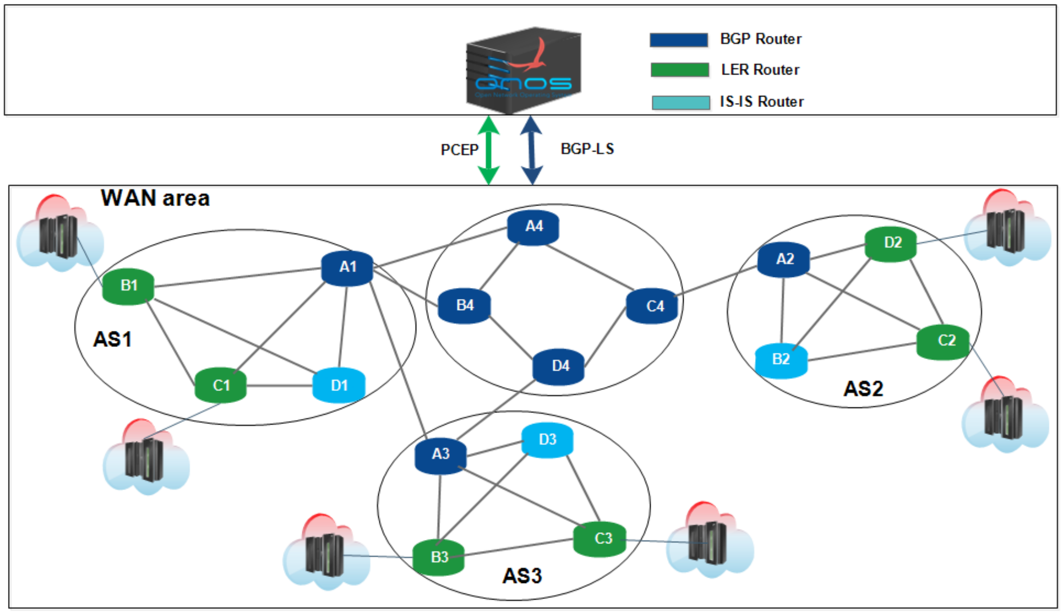

Fig. 3. Experimental platform topology

1) *Traffic Engineering Database (TED):* is a database which stores the traffic engineering information. TED is constructed using two main stores. First, link store managing the inventory of infrastructure links and holding links related to network layer reachability information (e.g., Link-ID, IP interface address, neighbors IP addresses, etc.) associated with links attributes describing network metrics (e.g., unreserved/max/available bandwidth, etc.). Such attributes can be easily gathered from the extensions of the IGP protocol. Second, device store describing the information related to routers (AS-ID, IGP router-ID) associated with node attributes such as: Router-ID, Node Name Multi-Topology identifier, etc.
2) *Link state Database (LS-DB):* is a database which stores the state of all the computed paths and their resources. This store is managed by our stateful Enhanced-PCE application.
3) *Tunnel manager:* is responsible for the creation, update and remove of tunnels. Tunnel manager takes as input the computed path communicated by Enhanced-PCE and the installed / updated / removed tunnel is stored. It is worth noting that the tunnel management and the calculation logics are dissociated which offers a high level of agility and extensibility.

The application level is built using two main components:

1) *Segment routing:* is responsible for the reservation and the allocation of labels. Indeed, a path is defined as a set of "segments" which can either be advertised by an IGP protocol or by using our enhanced-PCE application. Such a centralized approach offers the capability to tunnel services without the need of any signaling protocols such as RSVP-TE [5] or LDP [17]. Therefore, the architecture significantly scales more while simplifying the hardware requirements for the core routers.
2) *Enhanced-PCE application:* is responsible for computing a network path based on collected information in TED while taking into account specified computational constraints. Moreover, it exposes a REST interface through which user may specifies in a "Json" file the information related to its request. The latter is classified into two main groups:
    - Information related to the tunnel specification such as sourceID, destinationID, type of tunnel.
    - Information specifying the requirements in terms of bandwidth, latency, error rate, cost metric (i.e., IGP or TE).

III. SD-WAN PLATFORM

Our experimental platform relies into two main building blocks: i) emulated WAN infrastructure and ii) extended version of the ONOS SDN controller.

The WAN infrastructure is composed by IOS XRv router images [18] supporting BGP-LS protocol and running upon Qemu-KVM [19] virtualization technology. All the routers are running OSPF-TE as IGP. Moreover, BGP-LS, PCEP and segment routing are configured such as the peering between BGP routers and the ONOS SDN controller is operational. The network topology, as illustrated in Fig. 3, is composed by 4 Autonomous Systems ($\mathcal{AS}$s) interconnecting 6 data centers. It is worth noting that A4, B4, C4 and D4 routers belong to the Transit $\mathcal{AS}$ and consequently are label switching routers. However, $\mathcal{AS}_2$, $\mathcal{AS}_2$ and $\mathcal{AS}_3$ hold both label edge and label switching routers. The latter are connected to virtual machines running a Debian OS and playing the role of traffic generators.

To assess the effectiveness of our SD-WAN framework, we resolved the routing problem (named Enhanced PCE) in CPLEX solver within ONOS controller. Then, we compared

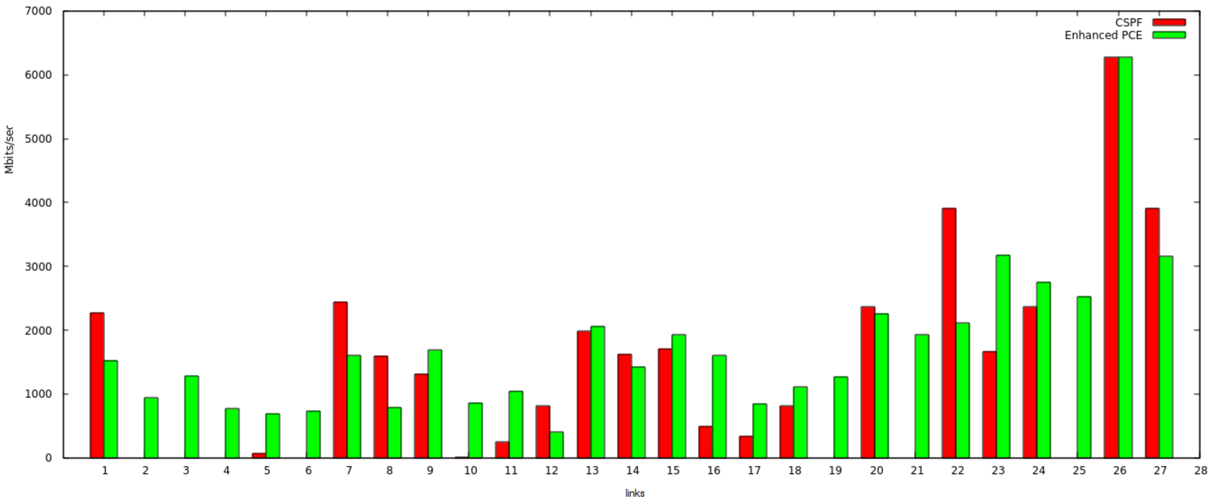

Fig. 4. Average links usage

the achieved load balancing with the frequently used protocol: Constraint Shortest Path First (`CSPF`).

Fig. 4 shows the circulating traffic in the different physical links of the infrastructure. We notice, that our SD-WAN achieves a high level of load balancing since the traffic is equitably shared out between the links. However, `CSPF` leads to a certain level of imbalance due to the fact that it selects always the shortest paths responding the demands.

IV. CONCLUSION

In this paper, we design an SD-WAN architecture to enhance the network resources allocation and hence improve the QoS of network sensitive applications. Our solution relies on ONOS controller and makes use of BGP-LS, PCEP and segment routing protocols in order to ensure a forward compatibility. As an extension of this work, we will optimize the allocation of paths in order to respond to the application requirements while ensuring the load balancing of the network. We evaluated through an experimental platform the performance of our proposed solution in terms of accepted bandwidth, loss rate and satisfaction level. Results show that our approach outperforms the Constraint Shortest Path First protocol.

ACKNOWLEDGMENTS

The authors would like to thank all the research engineers in Orange Labs for their support in the implementation of the testbed with ONOS controller and the allocated resources (processing power and memory) in the data center.

REFERENCES

[1] R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, “Md2-nfv: The case for multi-domain distributed network functions virtualization,” *2015 International Conference and Workshops on Networked Systems (NetSys)*, vol. 00, no. undefined, pp. 1–5, 2015.

[2] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, “Cloud RAN for mobile networks - A technology overview,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[3] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, “Fog computing: Principles, architectures, and applications,” *CoRR*, vol. abs/1601.02752, 2016.

[4] T. N. C. Srinivasan, A. Viswanathan, “Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base (MIB).” *RFC 3812*, 2000.

[5] D. G. T. L. V. S. G. S. D. Awduche, L. Berger, “RSVP-TE: Extensions to RSVP for LSP Tunnels,” *RFC 3209*, 2001.

[6] D. Thaler, “Multipath issues in unicast and multicast next-hop selection. internet engineering task force,” *RFC 2991*, 2000.

[7] H. Farhady, H. Lee, and A. Nakao, “Software-defined networking: A survey,” *Computer Networks*, vol. 81, pp. 79–95, 2015.

[8] J. Chen, X. Zheng, and C. Rong, “Survey on software-defined networking,” in *International Conference on Cloud Computing and Big Data in Asia*. Springer, 2015, pp. 115–124.

[9] M. H. Raza, S. C. Sivakumar, A. Nafarieh, and B. Robertson, “A comparison of software defined network (sdn) implementation strategies,” *Procedia Computer Science*, vol. 32, pp. 1050–1055, 2014.

[10] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow *et al.*, “ONOS: towards an open, distributed SDN OS,” in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 1–6.

[11] G. d. D. O and al., “First multi-partner demonstration of bgp-ls enabled inter-domain eon control with h-pce,” *Optical Fiber Communications Conference*, 2015.

[12] P. Aguilar Cabadas, “Pce prototype with segment routing and bgpls support,” 2014.

[13] “Introduction to segment routing,” http://packetpushers.net/introduction-to-segment-routing/, accessed: 2016-07-30.

[14] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, “North-bound distribution of link-state and traffic engineering (te) information using bgp,” Tech. Rep., 2016.

[15] D. Y. D. Katz, K. Kompella, “Traffic Engineering (TE) Extensions to OSPF Version 2,” *RFC 3630*, 2003.

[16] “PCE and PCEP Overview,” http://packetpushers.net/pce-pcep-overview/, accessed: 2016-07-08.

[17] J. L. J. Cucchiara, H. Sjostrand, “Definitions of Managed Objects for the Multiprotocol Label switching (MPLS), Label Distribution Protocol (LDP),” *RFC 3815*, 2004.

[18] A. Headquarters, “Cisco IOS XRv Router Overview,” 2015.

[19] E. KVM, “Kernel Virtual Machine,” 2016.