Conference Paper

Placement of Controllers in Software Defined Networking under Multiple Controller Mapping

Mohammad Ashrafi, Faroq AL-Tam, and Noelia Correia

CEOT, University of Algarve, Faro, Portugal

Abstract

This work focuses on the placement of controllers in software-defined networking architectures. A mathematical model is developed to place controllers under multi-controller switch-controller mapping, where a switch can be assigned to multiple controllers. Resiliency, scalability, and inter-plane latency are all modeled in the proposed model. A scalability factor is introduced to increase the load to capacity gap at controllers, preventing controllers to work near their capacity limit. The proposed model is shown to be effective and resilient under different failure scenarios while, at the same time, taking latency and scalability into consideration.

Keywords: Controller Placement, Software-defined Networking, Reliability, Scalability

Corresponding Author:
Mohammad Ashrafi
a41301@ualg.pt

Received: 26 November 2019
Accepted: 13 May 2020
Published: 2 June 2020

Publishing services provided by Knowledge E

1. Introduction

Software-defined Networking (SDN) is a software-oriented network design paradigm that allows network management to be simplified by decoupling the control logic from forwarding devices [1]. The control logic is the brain of the network that decides for best paths, prevents the outage and enforces the business logic into action. This is done in a centralized manner with the help of a computational device (server) instead of using the old distributed design, where every node in the network has the role of finding for best paths or recovering from failures. Besides the control plane, there is a data plane and a management plane. The control plane interacts with the data plane using southbound interfaces like OpenFlow [2]. Data plane is mostly responsible of forwarding data packets in switching or routing manner through programmable devices. The network policies are defined in the management plane, which communicates with the control plane via northbound interfaces.

To handle a received packet, switches consult the forwarding rules in their flow table(s). The packet is forwarded when a match is found; otherwise the switch sends a packet-in message to the master controller. Reactively, the controller determines the path for this packet and installs new rules in the affected switches. The time from receiving the packet-in message until a new response is received from the controller is called the flow setup time. Reducing this time is crucial for network efficiency, and this is basically influenced by the load at controllers and the propagation delay.

A Single controller system, like NOX [3] and Beacon [4], provides a holistic network-wide view but it represents a single point of failure and it lacks both reliability and scalability [5]. Therefore, multi-controller SDNs were developed, allowing the control plane to be physically distributed while keeping it logically centralized by synchronizing the network state among controllers [6]. This is the case of OpenDaylight [7] and Kandoo [8]. Multi-controller SDNs bring, however, new challenges like network synchronization [6] and adaptive switch- controller mappings [9].

A fundamental problem in multi-controller SDNs is the Controller Placement Problem (CPP) [10]. This problem focuses on deciding for the number of required controllers to meet the service level agreement for a network and where to find suitable locations for each controller. That is, to partition the network into subregions (domains), while taking into consideration some quality criteria and cost constraints [11, 12].

The CPP model discussed in this article incorporates the previously mentioned flow setup time, while presenting reliable and scalable solutions. Contrarily to the work in [13], where the authors make failure planning in a way to choose backup controllers and minimize two latencies, the latency from switch to the primary controller and to the backup controller, the approach followed in our work uses link failure scenarios and then decides for a controller placement that minimizes latency taking all failure scenarios into consideration. Besides such difference regarding protection, the scalability is also taken into account.

The remaining part of this paper is organized as follows. Section 2 discusses work related to CPP in SDNs. Section 3 discusses the mathematical model, whose results are discussed in Section 4. Finally, some conclusions are drawn in Section 5.

2. Related Work

A first work on controller placement is [10] where the goal is to minimize the inter-plane propagation delay. This is extended in [14] to incorporate the capacity of the controllers, since processing packet-in messages can causes overloading in controllers. In [15], in order to en- sure reliability, a metric called expected percentage of control path loss is proposed. The expansion case is considered in [16]. These articles usually model these problems as an integer-programming problem.

Regarding heuristic approaches to deal with this problem, the following exist. In [17] an approach is proposed that incorporates also switch migration. In [18], a QoS-aware CPP is ad- dressed and solved using a greedy and network partitioning algorithms. More recently, in [12] scalability and reliability issues in large-scale networks are considered, and clustering and genetic approaches are proposed in [19].

In [20], we have addressed the placement problem in a way that the flow setup time, reliability and scalability are considered. However, this first work fits into single controller mapping and while here, in this article, multiple mapping [9] is assumed. Although the goal will be the same, the set of constraints will be different due to the possibility of assigning multiple controllers to a single switch. Multiple controllers assignments to a switch is relatively new and will require some adaptation for OpenFlow, but it has been shown that, it can provide better properties, especially for resilience and load balancing problems, see [9, 21, 22]

3. Mathematical Model

Let us define $\mathscr{G}(\mathscr{N}, \mathscr{L})$ as the physical topology graph, where $\mathscr{N}$ is a set of physical nodes/locations and $\mathscr{L}$ is a set of physical links. Table 1 shows all notation used for known information, while Table 2 presents all required variables.

Table 1: Known information.

Term	Description
$\mathscr{C}$	Set of controllers.
$\mathscr{S}$	Set of switches
$\mathscr{N}_c$	Possible places for controller $c \in \mathscr{C}$, $\mathscr{N}_c \subseteq \mathscr{N}$
h_c	Number of requests per second that can be handled by controller $c \in \mathscr{C}$, i.e. controller capacity.
p_s	Number of requests not matching the flow table of $s \in \mathscr{S}$
$\mathscr{F}$	Set of physical link failure scenarios. Includes a scenario where all links are up.
$\mathscr{L}_f$	Set of physical links failing when scenario $f \in \mathscr{F}$ occurs.

3.1. Objective Function

$$\text{Minimize: } \delta + K_1 \frac{\Theta^{\text{Max}}}{\Delta} + K_2 \frac{\Pi^{\text{Max}}}{\Delta} \tag{1}$$

where Δ is a big value. The primary goal is to minimize the used percentage of controller capacity, δ, while reducing latency is a secondary goal. In this secondary goal, K_1 and K_2

TABLE 2: Required variables.

Variable	Description
σ_n^c	One if controller $c\in\mathscr{C}$ is placed at location $n\in\mathscr{N}_c$.
$\mu_{n_i,n_j,l,f}^{c_i,c_j}$	One if link $l\in\mathscr{L}\backslash\mathscr{L}_f$ is used for inter-controller $c_i - c_j$ communication, when controllers are placed at n_i and n_j, respectively, and failure $f\in\mathscr{F}$ occurs.
B_f^l	One if link $l\in\mathscr{L}\backslash\mathscr{L}_f$ is used for inter-controller communication when failure $f\in\mathscr{F}$ occurs.
$\gamma_f^{s,c}$	Fraction of flow from switch $s\in\mathscr{S}$ to controller $c\in\mathscr{C}$ when failure $f\in\mathscr{F}$ occurs.
$\phi_{f,l}^{s,c}$	Fraction of flow from switch $s\in\mathscr{S}$ to controller $c\in\mathscr{C}$, flowing at link $l\in\mathscr{L}\backslash\mathscr{L}_f$ when failure $f\in\mathscr{F}$ occurs.
$\omega_{s,f}^{c,n}$	Fraction of flow arriving to controller $c\in\mathscr{C}$, if located at node $n\in\mathscr{N}$, coming from switch $s\in\mathscr{S}$ when failure $f\in\mathscr{F}$ occurs.
δ	Used percentage of controller capacity.
Θ^{MAX}	Highest switch-controller latency, under any failure scenario.
Π^{MAX}	Highest number of links used for inter controller communication, under any failure scenario.

should be adapted according to switch-controller and inter-controller latency relevance. This objective function ensures, therefore, the linear scale up of the SDN network.

3.2. Constraints

This additional set of constraints must be fulfilled.

– Placement of controllers:

$$\sum_{n\in\mathscr{N}} \sigma^{c_n} = 1, \forall c \in \mathscr{C} \tag{2}$$

$$\sum_{\{l\in\mathscr{L}\backslash\mathscr{L}_f : src(l)=n\}} \mu_{n_i,n_j,l,f}^{c_i,c_j} - \sum_{\{l\in\mathscr{L}\backslash\mathscr{L}_f : dst(l)=n\}} \mu_{n_i,n_j,l,f}^{c_i,c_j} = \begin{cases} \sigma_{n_i}^{c_i}, & if\ n = n_i \\ -\sigma_{n_j}^{c_j}, & if\ n = n_j, \forall c_i, c_j \in \mathscr{C}, \forall n_i \in \mathscr{N}_{c_i}, \\ 0, & otherwise \\ & , \forall n_j \in \mathscr{N}_{cj}, \forall f \in \mathscr{F}, \forall n \in \mathscr{N} \end{cases} \tag{3}$$

Constraints (2) ensure a single location for each controller $c\in\mathscr{C}$, while constraints (3) ensure that there will be a path between every pair of controllers, under any failure scenario and considering their location. For controllers to communicate just with their local neighbors, the paths from any controller towards all the other controllers, should share as many links as possible, which is ensured by the following constraints.

$$\beta_f^l \geq \mu_{n_i,n_j,l,f}^{c_i,c_j}, \forall c_i, c_j \in \mathscr{C}, \forall n_i \in \mathscr{N}_{c_i}, \forall n_j \in \mathscr{N}_{c_j},, \forall l \in \mathscr{L}, \forall f \in \mathscr{F} \tag{4}$$

$$\Pi^{\text{MAX}} \geq \frac{1}{2} \sum_{\forall l \in \mathscr{L}} \beta_f^l, \forall f \in \mathscr{F} \tag{5}$$

and Π^{MAX}, counting for the highest number of hops between any two controllers, should be included in the objective function.

– Switch to controller mapping:

$$\sum_{\{c \in \mathscr{C}\}} \gamma_f^{s,c} = 1, \forall s \in \mathscr{S}, \forall f \in \mathscr{F} \tag{6}$$

$$\sum_{\{s \in \mathscr{S}\}} \gamma_f^{s,c} \times p_s \leq h_c \times \delta, \forall c \in \mathscr{C}, \forall f \in \mathscr{F} \tag{7}$$

where constraints (6) ensure 100% delivery of switch requests to controllers, and constraints (7) avoid the overload of controllers, while ensuring scalability regarding future multi- mapping reassignments and/or switch migration (triggered to deal with load fluctuations) due to the inclusion of δ used by the objective function. Again, the multiple failure scenarios are taken into consideration.

– Switch-controller latency:

$$\sum_{\{l \in \mathscr{L} \backslash \mathscr{L}_f : src(l)=n\}} \phi_{l,f}^{s,c,n_i} - \sum_{\{l \in \mathscr{L} \backslash \mathscr{L}_f : dst(l)=n\}} \phi_{l,f}^{s,c,n_i} = \begin{cases} \omega_{s,f}^{c,n}, & if\, n = loc(s) \\ -\omega_{s,f}^{c,n}, & if\, n = n_i, \forall s \in \mathscr{S}, \forall c \in \mathscr{C}, \forall f \in \mathscr{F}, \\ 0, & otherwise \\ & , \forall n_i \in \mathscr{N}_c, \forall n \in \mathscr{N} \end{cases} \tag{8}$$

where loc(s) is the location of switch s. The $\omega_{s,f}^{c,n}$ variables are limited by:

$$\sum_{\{s \in \mathscr{S}\}} \omega_{s,f}^{c,n} \leq \sigma_n^c \times \Delta, \forall c \in \mathscr{C}, \forall n \in \mathscr{N}_c, \forall f \in \mathscr{F} \tag{9}$$

$$\sum_{\{n \in \mathscr{N}_c\}} \omega_{s,f}^{c,n} = \gamma_f^{s,c}, \forall s \in \mathscr{S}, \forall c \in \mathscr{C}_c, \forall f \in \mathscr{F} \tag{10}$$

Switch-controller communication is ensured, under any failure scenario, according to placement of controllers. The highest latency is obtained by:

$$\Theta^{\text{MAX}} \geq \sum_{\{l \in \mathscr{L}\}} \sum_{\{n_i \in \mathscr{N}_c\}} \phi_{l,f}^{s,c,n_i}, \forall s \in \mathscr{S}, \forall c \in \mathscr{C}, \forall f \in \mathscr{F} \tag{11}$$

The Θ^{MAX} is also included in the objective function (1) for latency minimization.

– Variables domains:

$$0 \leq \delta, \gamma_f^{s,c}, \phi_{f,l}^{s,c}, \omega_{s,f}^{c,n} \leq 1; \sigma_n^c, \mu_{f,l}^{c_i,c_j}, \beta_f^l \in \{0,1\}; \Theta^{\text{MAX}}, \Pi^{\text{MAX}} \in \mathscr{R}^+. \tag{12}$$

This model assumes that the physical layer is not disconnectable under any single physical link failure. The CPLEX1 optimizer has been used to solve the problem instances discussed later in Section 4.

4. Experimental Results

4.1. Scenario Setup

Randomly generated topologies using NetworkX [23] have been used to evaluate the proposed model. These are: Topology I has 21 nodes, 40 links and 4 possible locations for controllers. Topology II has 21 nodes, 25 links and 4 available locations for controllers. Topology III has 21 nodes, 25 links and 9 available locations for controllers. The input parameter values, used by the optimizer, are displayed in Table 3.

Table 3: Parameters.

Parameter	Value
K_1, K_2	0.5, 0.5
p_s, h_s	[40, 100]; [500, 600]

Two different failure cases were used to evaluate the model. A case relates to single link failure (no two links fail at the same time in each scenario), while the other relates to multiple link failure scenarios. In both cases, two percentages for affected links were tested.

More specifically, Case I: refers to single link failure scenarios, where $\cup_{f \in \mathscr{F}} \mathscr{L}_f$ affects a total of 5% (Case Ia) or 15% (Case Ib) of all the links. Case II: refers to two or more links failing simultaneously, in each failure scenario, where $\cup_{f \in \mathscr{F}} \mathscr{L}_f$ affects a total of 5% (Case IIa) or 15% (Case IIb) of all the links.

Please note that in Case Ia, 5% of the links may fail, but no two links fail at the same time, while in Case IIa there is a 50 % probability for any two of the failed links to go down at the same time, which may lead to failure scenarios where two or more links fail simultaneously. In Case Ib and Case IIb, 15% of the links may fail.

4.2. Analysis

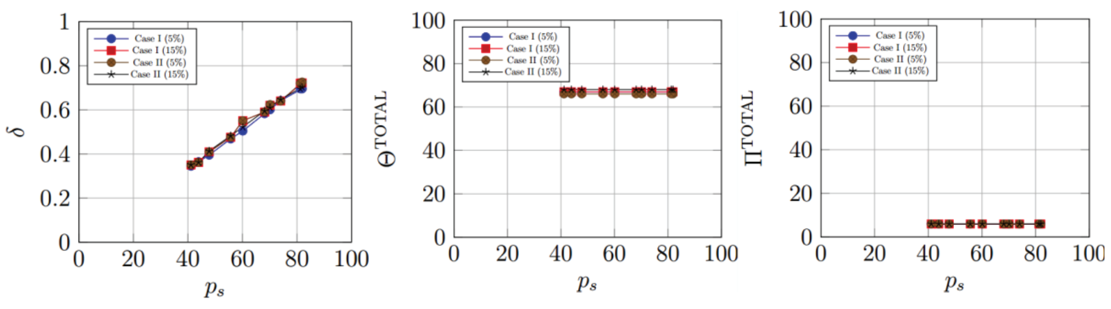
The experimental results of solving the proposed model for topologies I, II, and III, are shown in Figures 1, 2, and 3, respectively. The analysis of these plots is detailed next.

4.2.1. Scalability

The percentage of controller capacity utilization, δ, increases linearly as the number of packet-in messages increases for all topologies (Figures 1a, 2a, and 3a). This means that the gap between controller load and capacity is reducing due to increase of the received packet- in messages. For different failing scenarios the model presents similar results that is an indicator that the model is able to effectively deal with different failure scenarios, presenting similar scalability factors. This also indicates that the model is well behaved under increasing load.

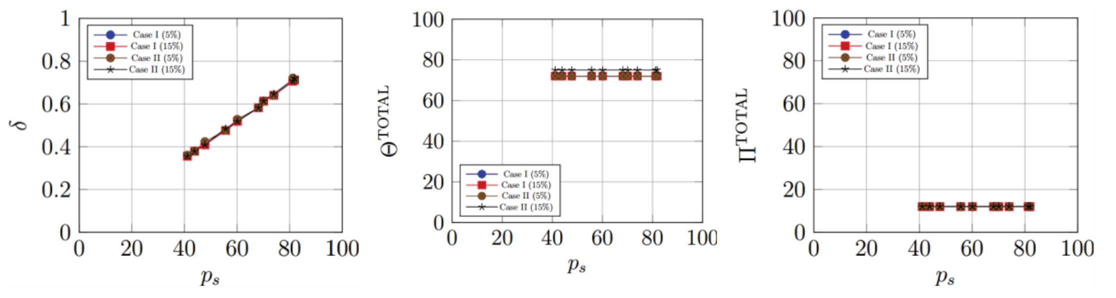
4.2.2. Latency

Concerning latency, the switch-controller delay of Topology I (Figure 1b) is clearly lower than the one in Topology II (Figure 2b). This is due to the fact that Topology I has more links than Topology II, which allows the model to find better locations for the controllers. Surprisingly, Topology III has slightly better switch-controller latency results (Figure 3b) than those of Topology I, which tells us that allowing more potential locations for the controller can actually compensate to the reduced density of the network. The inter-controller delays in Topology I (Figure 1c) are lower than those of other topologies, since there are more connections between the networking nodes.



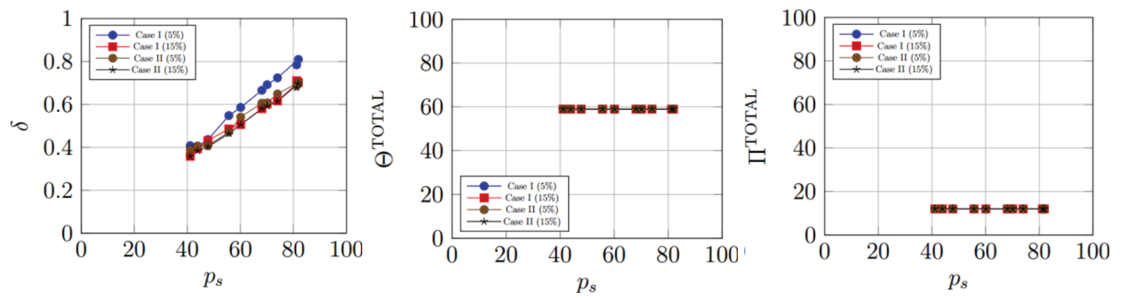
(a) Scalability factor. (b) Switch-controller latency. (c) Inter-controller latency.

Figure 1: Results for Topology I, having high connectivity and relatively low number of possible locations for the controllers.



(a) Scalability factor. (b) Switch-controller latency. (c) Inter-controller latency.

Figure 2: Results for Topology II, having relatively low connectivity and number of possible locations for the controllers.



(a) Scalability factor. (b) Switch-controller latency. (c) Inter-controller latency.

Figure 3: Results for Topology III, having relatively low connectivity and relatively high number of possible locations for the controllers.

5. Conclusions

It has been presented a mathematical model for solving the controller placement under multi-controller switch-controller assignment, where a switch can be mapped to multiple controllers. The proposed model is able to capture important qualities like scalability, resiliency, and inter-plane delay under various failure scenarios.

The results show that scalability is ensured by minimizing the used percentage of controller capacity, in order to easily handle any sudden change in the load. Considering the latency, the increase in inter-plane delay can be compensated through adding more freedom in controller's locations. This way, the model is able to selectively find optimal positions for controllers such that the overall latency is minimized. The model is also shown to behave nicely under multiple failure scenarios, presenting similar results for more critical failure scenarios and less critical ones. Regarding the density of the network, the proposed model is also able to work properly regardless of network connectivity and, it turns out that the network density can be compensated when adding more freedom to controller's potential positions.

As a future work, we will consider extending our model to support different qualities like cost, potential upgrade, and network function virtualization.

Acknowledgments

This work was supported by FCT (Foundation for Science and Technology) from Portugal within CEOT (Center for Electronic, Optoelectronic and Telecommunications) and UID/MULTI/00631/2019 project.

References

[1] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. 2008. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69–74.

[2] Foundation, O. N. 2011. Openflow switch specification. Technical report.

[3] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. 2008. Nox: Towards an operating system for networks. SIGCOMM Comput. Commun. Rev., 38(3):105–110.

[4] Erickson, D. 2013. The beacon openflow controller. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN '13, pages 13–18, New York, NY, USA. ACM.

[5] Yeganeh, S. H., Tootoonchian, A., and Ganjali, Y. 2013. On Scalability of Software-Defined Networking. IEEE Communications Magazine, 51(2):136–141.

[6] Zhang, Y., Cui, L., Wang, W., and Zhang, Y. 2018. A survey on software defined networking with multiple controllers. Journal of Network and Computer Applications, 103:101 – 118.

[7] Medved, J., Tkacik, A., Varga, R., and Gray, K. 2014. OpenDaylight: Towards a Model-Driven SDN Controller Architecture. In 2014 IEEE 15th Inter Symposium On A World Of Wireless, Mobile And Multimedia Networks (WOWMOM).

[8] Hassas Yeganeh, S. and Ganjali, Y. 2012. Kandoo: A framework for efficient and scalable offloading of control applications. In Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN '12, pages 19–24, New York, NY, USA. ACM.

[9] AL-Tam, F. and Correia, N. 2019. Fractional switch migration in multi-controller software- defined networking. Computer Networks, 157:1 – 10.

[10] Heller, B., Sherwood, R., and McKeown, N. 2012. The Controller Placement Problem. ACM SIGCOMM Computer Communication Review, 42(4):473–478.

[11] Sallahi, A. and St-Hilaire, M. 2015. Optimal Model for the Controller Placement Problem in Software Defined Networks. IEEE Communications Letters, 19(1):30–33.

[12] Bannour, F., Souihi, S., and Mellouk, A. 2017. Scalability and reliability aware sdn con- troller placement strategies. In 2017 13th Inter. Conf. on Network and Service Management (CNSM), pages 1–4.

[13] Killi, B. P. R. and Rao, S. V. 2016. Controller placement with planning for failures in software defined networks. In 2016 IEEE Inter. Conf. on Advanced Networks and Telecom- munications Systems (ANTS), pages 1–6.

[14] Yao, G., Bi, J., Li, Y., and Guo, L. 2014. On the Capacitated Controller Placement Problem in Software Defined Networks. IEEE Communications Letters, 18(8):1339–1342.

[15] Yannan, H., Wendong, W., Xiangyang, G., Xirong, Q., and Shiduan, C. (2014). On Reliability-optimized Controller Placement for Software-Defined Networks. China Communi- cations, 11(2):38–54.

[16] Sallahi, A. and St-Hilaire, M. 2017. Expansion Model for the Controller Placement Problem in Software Defined Networks. IEEE Communications Letters, 21(2):274–277.

[17] Yao, L., Hong, P., Zhang, W., Li, J., and Ni, D. 2015. Controller Placement and Flow based Dynamic Management Problem towards SDN. In 2015 IEEE Inter. Conf. on Communication Workshop (ICCW), pages 363–368.

[18] Cheng, T. Y., Wang, M., and Jia, X. 2015. QoS-Guaranteed Controller Placement in SDN. In IEEE Global Communications Conference (GLOBECOM).

[19] Wang, G., Zhao, Y., Huang, J., and Wu, Y. 2018. An Effective Approach to Controller Placement in Software Defined Wide Area Networks. IEEE Trans. On Network And Service Management, 15(1):344–355.

[20] Ashrafi, M., Correia, N., and Al-Tam, F. 2018. A scalable and reliable model for the place- ment of controllers in SDN networks. In Broadband Communications, Networks, and Systems - 9th Inter. Conf., BROADNETS 2018, Faro, Portugal, September 19-20, 2018, Proceedings, pages 72–82.

[21] V. Sridharan, G. Mohan, and T. Tram. “On multiple controller mapping in software defined networks with resilience constraints.” IEEE Communications Letters 21, no. 8 (2017): 1763- 1766.

[22] G. Burak, T. Sinan, T. Murat, C. Seyhan, and L. Erhan, “Dynamic Control Plane for SDN at Scale”, Inter. Journal on Sele. Areas in Comm. vol. 36, pp. 2688-2701, 2018.

[23] Aric, H., Pieter, S., and Daniel, S.C. 2008. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7thPython in Science Conference (SciPy2008), Pasadena, CA USA, Aug.2008, pp. 11-15.