Enduring Node Failures through Resilient Controller Placement for Software Defined Networks

Maryam Tanha, Dawood Sajjadi, and Jianping Pan
Department of Computer Science, University of Victoria, Victoria, BC, Canada

***Abstract*—Software Defined Networking (SDN) is an emerging paradigm for network design and management. By providing network programmability and separation of control and data planes, SDN offers salient features such as simplified and centralized management and control, reduced complexity and accelerated innovation. However, SDN introduces new challenges that should be addressed properly in order to benefit from its unprecedented capabilities. Due to the (logically) centralized control in SDN, the resilience of the control plane has a great impact on the functioning of the whole system. In this case, resilient controller placement problem (how many controllers are needed and where to place them to provide higher reliability) is a hot research topic that affects the reliability and performance of SDN in Wide Area Networks (WANs). Thus, we define a resilient controller placement problem, which satisfies a set of constraints, some of which are missing in the existing solutions. The acquired results on real tier-1 US service provider network topologies demonstrate the effectiveness of the approach. This can give helpful insights to the network operators for designing or modifying their network topologies to enhance the resilience of the control plane in SDN.**

I. Introduction

Today, the evolution of the Internet's physical infrastructure, its protocols and performance, has become extremely demanding due to its rapid growth and large-scale deployments [1]. Moreover, emerging Internet applications such as Internet of Things (IoT), Cloud Computing and Big Data underscore the need for faster, more scalable, efficient, secure and resilient network architectures. Software Defined Networking (SDN) paradigm, which involves the decoupling of the data and control planes of a network, shows promises to be the next generation of the networking architectures. Using SDN, the control and management of network devices are performed by centralized software, called controllers. SDN contributes towards facilitated and efficient network design, management and control by providing vendor-independent control interfaces [2], [3].

In spite of the great virtues of utilizing SDN solutions, there are some open issues and challenges that should be addressed properly to benefit from the unprecedented features of SDN. More specifically, due to the fact that the controllers are the heart of SDN functionality, they are the main resilience bottlenecks. The controllers may fail randomly resulting from natural disasters, power outage, security bugs, and malicious/terrorist attacks. These disruptive events may cause devastating impacts on the network infrastructure by (partially) demolishing controller instances in a geographical area and subsequently affecting many network applications and services. Therefore, the SDN control plane requires high level of resilience which is tightly interwoven with the controller placement problem. The controller placement indicates the number of required controllers to handle the switches' demands as well as their location (in the network topology) in an efficient and cost-effective manner.

Controller placement affects almost all of the resilience disciplines, including survivability (as a superset of fault tolerance), dependability (as a superset of reliability), security, performability, traffic tolerance, and disruption tolerance. An investigation of the key research efforts addressing the resilience disciplines in the context of SDN can be found in [4]. It should be noted that some of the offered solutions can address more than one resilience discipline since there are overlaps among the applied methods (e.g., redundancy improves both fault tolerance and reliability).

In this paper, we use the umbrella term "resilient controller placement" to refer to our proposed controller placement problem, since it covers more than one resilience discipline. Particularly, we focus on the controllers' failures and define the resilient controller placement problem while taking into account important factors (some of which are missing in existing works) such as flow setup latency, the incurred load by switches and the capacity of the controllers. It should be noted that the terms "incurred load by the switches" and "switches' demands" are used interchangeably in this paper.

The contributions of this research are twofold. First, we present a resilient control plane design and formulation by having physically distributed and redundant controllers. Thus, a switch can be managed by different controllers at different resilience levels. Also, the capacity of the controllers, the incurred load by switches on the controllers, and the propagation latency between the switches and their assigned controllers are taken into account. Second, we apply the proposed model and formulation to real US tier-1 network topologies and give helpful insights into the design of a resilient SDN by analyzing and comparing the controller placements for different service providers. We believe that the presented comparisons among the network topologies, which cover almost the same geographical regions, help the network operators to choose a topology or amend their existing designs to reduce the overhead of the resilient controller placement.

The rest of the paper is organized as follows: Section II gives an overview of the existing works on controller placement in SDN. The problem statement as well as the proposed formulation is presented in Section III. Section IV analyzes

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

the performance of the proposed solution using real network topologies by considering different metrics. Finally, Section V concludes the paper.

II. Related Work

In this section, we review some of the existing works on the controller placement in SDN by putting special emphasis on the resilient controller placement solutions.

A. Controller Placement

One of the most significant factors to place the controllers in an SDN-based WAN is the round-trip propagation latency, which can be calculated by using the length of the shortest path between a switch and its assigned controller(s) [5], [6]. Thus, many research works have approached the controller placement problem mainly from this aspect. The authors in [5], introduced latency-related metrics corresponding to variants of the Facility Location Problem (FLP). Then, they assessed the impact of controller placements on different publicly available WAN topologies. The outcome of their experiments demonstrated that one controller is sufficient to satisfy the latency requirements. In addition, although the method was exact, it had high cost regarding the CPU time and exponential complexity with respect to the optimal number of placements. The capacitated controller placement problem (as a variant of capacitated k-center problem) was investigated in [6]. It involved positioning the controllers to minimize the propagation latency while taking into account the capacity of the controllers and demand of the switches. The authors provided a modified version of an exact algorithm for capacitated k-center problem, which efficiently reduced the number of required controllers to prevent from controller overload, and evaluated it on real WAN topologies.

An optimal model for the controller placement in small scale SDNs was proposed in [7]. The offered linear programming model was solved by the CPLEX optimizer and indicated the optimal number, the type of the controllers, and their locations along with the interactions among the controllers assuming a switch can be managed by more than one controller. However, the solution was limited by its high time-complexity and it was not evaluated on real network topologies. The formulated controller placement problem in [8], sought the minimum cost of controller deployment by considering the delay from the switches to their assigned controllers as well as the weights of the switches (with regard to their importance).

The controller placement, as a principal deployment aspect of a proposed decentralized management and control framework, was formulated as an uncapacitated FLP in [9]. The authors presented an algorithm to determine the configuration of the distributed management and control planes. The open-source Pareto-based Optimal COntroller (POCO) placement framework provides pareto-optimal placements with regard to different measures, including switch-controller latency, controller-controller latency and controller load imbalance. Using heuristics, this framework assists network operators with planning controller placements in large scale or dynamic networks, and evaluating the trade-off between accuracy and time-constraints. One important downside of this work is not involving the capacity of the controllers and the incurred load on the controller by the switches into the problem formulation.

B. Resilient Controller Placement

As mentioned before, the resilience of the control plane plays a significant role in the resilience of the whole SDN network. The research carried out in [10]–[12] is mainly concerned with resilient controller placement to improve the resilience of the south-bound connections (controller-switch connections) in SDN, whereas in this paper we focus on the resilient controller placement to enhance the resilience of the control plane (dealing with controller node failures).

The authors in [13] studied the controller placement using the interdependent network analysis. They formulated the controller placement to improve a defined resilience metric and solved the problem using a greedy optimization method and partitioning on different types of network topologies (e.g., star, and ring). A controller placement strategy for improving the survivability in SDN was proposed in [14]. More specifically, three main aspects, including connectivity, capacity, and recovery were considered. The controller placement problem was formulated as an integer linear program to maximize connectivity while satisfying the controller capacity constraints. Also, a percentage as the backup capacity was set for each controller and two heuristic algorithms were proposed for defining the list of backup controllers after placing the controller instances.

One of the most recent research works on the reliability of controller placement in SDN has been conducted in [15] by extending an initial work in [16]. As a variant of the fault-tolerant FLP, the authors introduced the Fault Tolerant Controller Placement (FTCP) problem to achieve high south-bound reliability. In the proposed formulation, each switch is required to satisfy a reliability constraint in a way that the operational route to any of its connected controllers remains with at least a given probability. The simulation outcome of applying the proposed heuristic algorithm to several network topologies, demonstrated that being connected to two controllers suffices for each switch. Other aspects of the controller placement such as controller load were also investigated. Not incorporating the controllers' capacities and the demands of the switches into the problem formulation is the drawback of the proposed formulation. Moreover, the heuristic algorithm shows acceptable runtime; nevertheless, no comparison was made with the optimal solution.

III. Problem Statement and Formulation

A. Problem Description

The failures of the controllers (software or hardware failures) in an SDN-based network can be caused by natural disasters or intentional attacks. To achieve high resilience in the control plane, the controller placements should fulfill the fault tolerance requirements in addition to the performance and cost criteria. Using redundancy in assigning controllers to the switches is a well-advised method. Multiple backup

controllers can be assigned to a switch at different resilience levels (primary, secondary, and so on). Maintaining backup controllers may follow a specific replication protocol which involves inter-controller communication and synchronization or it may require that a list of backup controllers is incorporated into a switch. Considering the latter, we can define a resilient controller placement problem, as an optimization problem, to meet the resilience constraints as well as to address the performance (mostly related to the propagation latency as discussed in Section II), the cost and capacity limitations. The cost limitations are associated with the number of required controllers or having a budget in terms of the the number of controller instances, and the inherent cost of deployments (e.g., CAPEX and OPEX). Also, the capacity constraints assist in dealing with the load on a controller. More specifically, due to the resource constraints (CPU, memory, and access bandwidth) each controller can only manage a determined number of requests. If the controller becomes overloaded, the processing latency will go up and subsequently, affect the latency between a switch and the controller (it becomes a non-negligible part of the total latency). Moreover, overloaded controllers have a higher probability of failure [6].

B. Problem Formulation

The topology of an SDN-based network is represented as a connected graph $G(V = S \cup C, E)$, where V is the set of nodes (including the sets of OpenFlow-enabled switches, i.e., S, and potential controllers' sites, i.e., C) and E denotes the set of links. Assuming the controllers can be co-located with the switches, the potential locations for the controllers will be equal to the set of switches (i.e., $C = S$). Each switch s incurs a load l_s (the demand of the switch) on its assigned controller. This load mainly results from processing PACKET_IN events. Each controller c is associated with capacity Q_c. Now, we can define the Resilient Controller Placement (RCP) optimization problem as follows:

Minimize

$$\sum_{c \in C} f_c y_c + \sum_{s \in S} \sum_{c \in C} \sum_{r=0}^{m} l_s d_{sc} p_f^r (1 - p_f) x_{scr} \tag{1}$$

subject to,

$$\sum_{c \in C} x_{scr} = 1 \quad \forall s \in S, r = 0, ..., m \tag{2}$$

$$x_{scr} \leq y_c, \quad \forall s \in S, c \in C, r = 0, ..., m \tag{3}$$

$$\sum_{r=0}^{m} x_{scr} \leq 1 \quad \forall s \in S, c \in C \tag{4}$$

$$\sum_{s \in S} \sum_{r=0}^{m} l_s x_{scr} \leq Q_c \quad \forall c \in C \tag{5}$$

$$x_{scr} \in \{0, 1\} \quad \forall s \in S, \ \forall c \in C, r = 0, ..., m \tag{6}$$

$$y_c \in \{0, 1\} \quad \forall c \in C \tag{7}$$

In the above formulation, the cost f_c is associated with setting up (may include other costs according to the preference of the network operators) a controller at node c and d_{sc} is the shortest-path length (the link costs are the propagation delays) between switch s and the controller located at node c. When a controller failure occurs, the switches managed by the failed controller should be connected to the next backup controller in their list to minimize the network disruption. This would incur more cost; thus, the main objective is to minimize the switch-controller re-assignment costs after the possible controller(s)' failures. In RCP, r denotes the resilience level at which a controller serves a given switch. For instance, $r = 0$ indicates a primary assignment of a controller to a switch, and $r = 1$ denotes the assignment of the first backup controller to a switch. Therefore, if a switch s's level-r assigned controller fails, the level-$(r + 1)$ assigned controller would serve it as backup. We assume that all of the controllers have a uniform failure probability denoted by p_f and they fail independently from each other. Also, m denotes the maximum resilience level. Moreover, multiple failures may happen simultaneously. The binary decision variable y_c holds value 1 if node c (among the potential controller locations) is selected to deploy a controller and 0 otherwise. The level-r assignment of a switch s to the controller located at node c is denoted by setting a binary variable x_{sc} to 1.

The first part of the objective function computes the total cost for the deployment of the controllers. The second part shows the expected routing costs (to the switches) from the controllers. These two costs should be normalized and added together. Constraint (2) expresses that each switch s must be managed by a controller c at level r. Constraint (3) prohibits a switch from being assigned to a controller site which is not open while constraint (4) indicates that a controller can only serve a switch at one resilience level. Constraint (5) prevents the total incurred loads by the switches managed by a controller from exceeding the controller's capacity. It should be noted that each controller can manage different switches at different resilience levels. Constraints (6), (7) are the integrality constraints.

The RCP problem is similar to the Capacitated reliable Fixed-charge Location problem (CRFLP) [17], [18], in the sense that the controllers can play the role of the facilities and switches are the customers/clients. However, unlike the models proposed in [17], [18], we assume the upper bound m ($m << |C|$) for r, which is not necessarily equal to the cardinality of C. Moreover, the authors assumed an emergency facility that was never disrupted with zero value for f_c. The main usage of the mentioned facility was when all open facilities failed, or if the emergency cost was smaller than the

cost of serving a given customer from its kth-nearest facility ($k < r$) when the first $k-1$ facilities failed. The use of this emergency facility resulted in buying the products from a competitor on an emergency basis which does not apply to our scenario for controller placement.

IV. Performance Evaluation

A. Evaluation Setup

We solved the RCP optimization problem using the MATLAB API of the GUROBI optimization software (version 6.5.1) [19] on the network topologies of US continental tier-1 service providers [20] obtained from the Internet Topology Zoo [21]. Particularly, the chosen Point of Presence (PoP)-level topologies are Sprint, ATT North America (two maps, one before 2008 and another in 2008), PSINeT (now part of Cogent Communications), and UUNET (now part of Verizon Business). These tier-1 ISPs are among the major US ISPs and cover different US states. The associated PoP-level maps are interesting since they show different topologies that cover almost the same geographical region. Moreover, the PoP-level maps are important in network design optimizations and they are the level at which the resilience and redundancy are most likely to be considered [21]. Table I shows the information about the chosen network topologies. ATT NA (1) and ATT NA (2) denote the two versions of the ATT North America' map (as mentioned above), respectively. It should be noted that although the UUNET has some nodes located in Canada, the main part of the network resides in the US, which is sufficient for our purpose.

Using the information from the maps, we obtain the geographical locations of the nodes, and the links connecting them as well as we filter out very close nodes. Then, we perform the evaluation on the resultant connected graph. The cost f_c can be associated with the cost of deployment and maintenance or other economic costs for a controller instance. While part of this cost may be fixed for placing a controller at any potential location, the other part of it can be according to the preference of the network operators. In this paper, we focus on the latter and associate f_c with the node properties in the network topology graph. More specifically, the higher the degree of a potential node (location) for the controller, the less the f_c is. This mainly results from the fact that nodes with better connectivity are reachable from more nodes, and placing the controllers on such nodes most probably decreases the cost in terms of the number of controllers.

We run our experiments with $m = 0$ (no backup), $m = 1$ (i.e., $r = 0, 1$) and $m = 2$ (i.e., $r = 0, 1, 2$). The values of the switches' demands (as average values) and controllers' capacities are assigned based on the prior studies in [7], [14]. In addition, we assume in-band control, i.e., no dedicated links between controllers and switches for control traffic. Also, the shortest-path length between a switch and a controller is the sum of the propagation latencies of all the links along the path. We consider the following three scenarios and run 10 independent experiments for each of the mentioned maximum resilience levels for all of the topologies (note that increasing the number of experiments did not affect the results significantly).

1) In this scenario, we assume homogeneous switches (i.e., switches with homogeneous demands equal to 500 kilo req/s) and controllers (i.e., controllers with homogeneous capacities equal to $5,000$ kilo req/s). The probability of the node failure (i.e., p_f) is assumed to be randomly chosen from $[0.01\ 0.25]$ according to the studies of real failures in [22], [23].
2) To have a more realistic and practical scenario, we consider the population of the states where the switches are located as a rough estimation for the incurred loads of the switches on their assigned controllers. Particularly, since most of the topologies' network date is 2011 (from the Topology Zoo), we use the population of the states for that year [24]. The populations of the covered states in the maps are in $[500,000\ \ 38,000,000]$, and we group them into 18 clusters starting with $[500,000\ \ 2,000,000]$ and ending with $[36,000,000\ \ 38,000,000]$. The corresponding demands of the switches are configured with the minimum and maximum values of 150 kilo req/s and $1,000$ kilo req/s (with step size of 50), respectively. The capacities of the controllers and p_f are the same as the first scenario.
3) In this scenario, we assume heterogeneous values for switches' demands and the controllers' capacities. Specifically, the demands of the switches are uniformly distributed in $[200\ \ 1,000]$ kilo req/s and controllers' capacities have values in $[1,800\ \ 8,000]$ kilo req/s with fixed p_f as 0.05. Here, we relate the capacity of a controller to its location and the capacities are generated for all of the potential locations of the controllers at the initial phase. This is justified when we have a case that the resources of an ISP are limited at specific locations; thus, they impose a constraint on the capacity of a controller placed there.

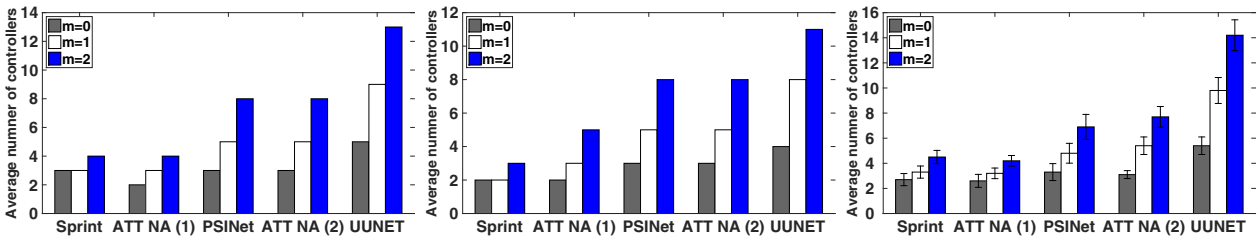
B. Numerical Results and Discussion

We analyze the results considering four aspects, namely the number of assigned controllers, the propagation latency between the switches and their assigned controllers, the controller locations at different resilience levels, and the distribution of the loads among the controllers. These results shed light on the trade-off between the resilience levels and other criteria such as the performance and cost.

*1) **The number of assigned controllers**:* Fig. 1 illustrates the average number of assigned controllers per scenario. As shown in the figure, increasing the resilience level results in a rise in the number of required controllers for each network topology. Particularly, the number of assigned controllers (as the percentage of network size) at the maximum resilience level ($m = 2$) for all of the topologies and the three scenarios are tabulated in Table II. It can be seen that having a higher resilience level is more cost-effective (in terms of the number of required controllers) for the UUNET which has larger network size as well as more redundant paths and higher

TABLE I: Information about the chosen network topologies.

	Sprint	ATT NA (1)	PSINet	ATT NA (2)	UUNET
Network size	11	12	24	25	42
Links	18	21	25	56	77
Average node degree	3.27	3.5	2.08	4.48	3.6



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 1: Average number of the required controllers in each topology for all the three scenarios.

TABLE II: Required number of controllers as a percentage of network size for $m = 2$.

Topology	Scenario 1	Scenario 2	Scenario 3
Sprint	36%	27%	40%
ATT NA (1)	33%	41%	35%
PSINet	33%	33%	28%
ATT NA (2)	32%	32%	30%
UUNET	30%	26%	33%

average node degree. Also, we can see that the aforementioned cost has been reduced for the ATT NA (2) compared with ATT NA (1) (example of a network at two different stages). Thus, although all of the topologies nearly cover the same geographical region with some common locations for the nodes, the number of the required controllers is topology-dependent. It should be noted that regardless of the assigned probability of failure to the nodes, the number of required controllers are quite similar in the first and second scenarios.

As shown in Fig. 1a and Fig. 1b, the average number of required controllers at different resilience levels and for different network topologies in scenario 1 are higher than their peers in scenario 2. This results from the constant demands of the switches in the former compared with variable demands in the latter (the total demands are less than scenario 1), which requires fewer controllers to accommodate the incurred loads of the switches. Considering the average number of assigned controllers in the third scenario, as illustrated in Fig. 1c, the acquired results show acceptable standard deviations in different experiments with random values for the switches' demands and the controllers' capacities. Also, they demonstrate similar trends to the obtained results in the first and second scenarios. In addition, the results show that the number of required controllers not only relies on the resilience level, but also depends on the topology and the given scenario (e.g., having the same number of controllers for Sprint in Fig. 1a).

*2) **Propagation latency**:* Propagation latency as the main contributor to the flow-setup latency is the focus of this section. To obtain the propagation latency, the distance between each two nodes is calculated using the spherical law of cosines as an approximation of the air-line distance (as a rough estimation without knowing the fiber lengths) [25] and divided by the signal speed. The propagation latency threshold (on the shortest path) from a switch to its assigned controller at any resilience level is assumed to be 250 ms [7].

Fig. 2 demonstrates the Cumulative Distribution Function (CDF) of the propagation latency at each resilience level per tier-1 topology for the second and third scenarios (more complex scenarios). Considering the highest resilience level (i.e., $m = 2$), the maximum propagation latencies for all of the topologies are below 50 ms, which is far less than the latency threshold and leaves the room for other contributors of the flow-setup latency, including transmission delay, processing delay and probably the delay incurred by congestion in the network. In most cases, when the resilience level is increased (e.g., from $m = 0$ to $m = 1$), the maximum latency between a switch and its assigned controller in a topology goes up. But this is not always the case; for instance, as shown in Fig. 3 for scenario 3 and ATT NA (2), the maximum latency decreases when the resilience level increases from 1 to 2 in the second experiment. This is mainly due to the change in the placement of the controllers when changing the resilience level. Other factors are the increase in the number of assigned controllers by rising the resilience level as well as the capacities of the controllers. In this experiment, the IDs of the assigned controllers are $\{1, 3, 6, 14, 18\}$ for $m = 1$, whereas the controller IDs are $\{3, 6, 7, 9, 14, 17, 18\}$ for $m = 2$. The former includes node ID 1 which incurs the maximum latency (43 ms) between itself and one of its controlled switches (i.e., node ID 20). While in the latter, the same switch (node ID 20) is managed by a controller with node ID 17 and the latency is reduced to 17 ms. Similar explanations apply to the other experiments in the same figure.

Furthermore, considering ATT NA (1) and ATT NA (2) in Fig. 2, the maximum propagation latency between a switch and its assigned controller has been decreased from around 40 ms to less than 30 ms. This result for latency is in line with the results with regard to the number of controllers. Both

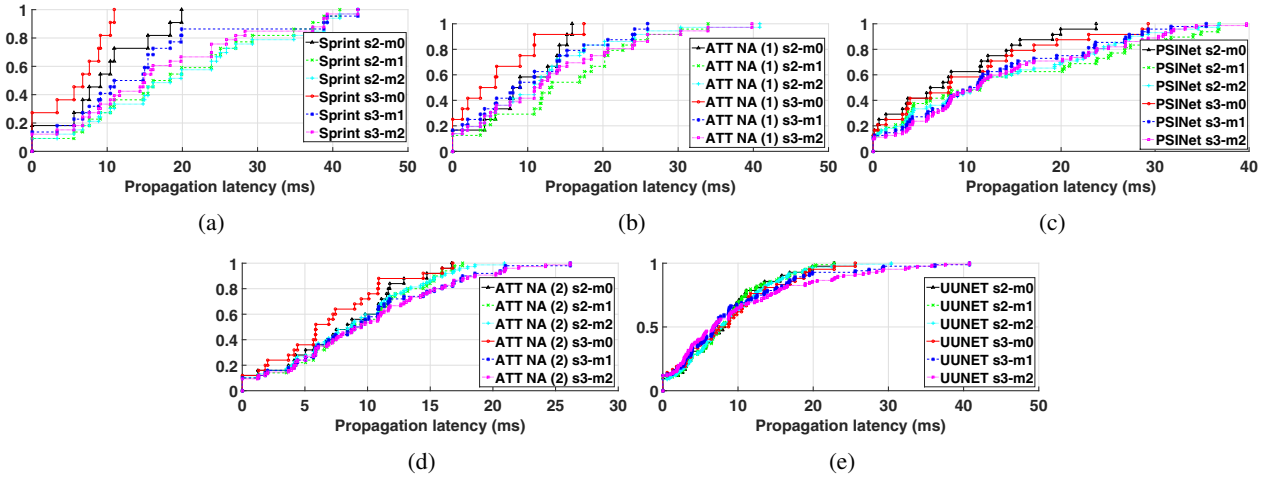


Fig. 2: CDF of the propagation latency for all the topologies in scenarios 2 and 3.

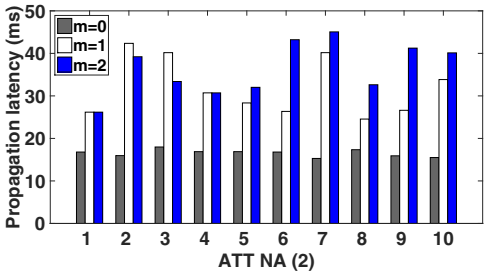


Fig. 3: Maximum propagation latency for scenario 3 (ATT NA (2)).

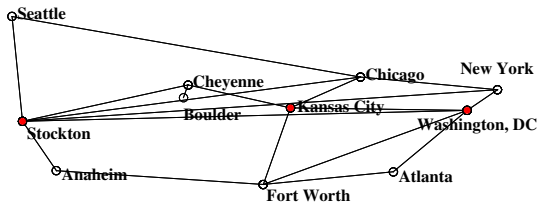


Fig. 4: Controller locations for Sprint with $m = 2$ in scenario 2.

costs in terms of the number of required controllers and the latency between the switch and controller have been decreased in the updated topology (i.e., ATT NA (2)).

*3) **Controller locations**:* As mentioned earlier, the chosen tier-1 ISPs cover almost the same geographical area and they have some nodes in relatively close geographical regions. As a representative example to show the locations of the controllers at a resilience level, we choose the Sprint topology and depict the locations in Fig. 4 for the second scenario. The red nodes indicate the assigned controllers. As shown in this figure, Stockton, Kansas City, and Washington, DC are the nodes, where the controllers are located. One reason is due to their higher connectivity (higher node degree) and subsequently, better reachability from other nodes. This is reflected by our assumed location-dependent deployment cost (f_c) which increases in inverse proportion to the node degree.

Another interesting insight from the acquired results for the controller locations is that the chosen topologies have some controller locations in common. As an example, for $m = 2$, Kansas City as a controller location, is common among Sprint, PSINet, and UUNET in all of the scenarios and experiments. Similarly, Los Angeles serves as a controller location at resilience level $m = 2$ for ATT NA (1), ATT NA (2), and UUNET in all of the three scenarios. Both Kansas City and Los Angeles are among the nodes with higher average node degree. Also, Kansas City has a strategic location in most of topologies since it connects east and west of the US in the maps. Moreover, Los Angeles is located in the state with the highest population. Thus, these cities are more attractive for the network operators to have PoP sites and subsequently controllers there. There is also the possibility of sharing the infrastructure at these locations for the service providers.

*4) **Load distributions among the controllers**:* Table III presents the distribution of the load in terms of the number of switches managed by a controller at different resilience levels. Each 4-tuple shows the minimum, maximum, mean and standard deviation for the number of switches managed by the controllers in a given topology. The difference between the maximum and minimum values called the load imbalance (a lower value is desirable). While increasing the resilience level results in more load imbalance for some topologies in different scenarios, it leads to less load imbalance for the others. This again confirms the reliance of the solution on the network topology. As shown in the table, increasing the resiliency level for Sprint (from $m = 1$ to $m = 2$) in scenario 2 does not affect the load imbalance. Also, as illustrated in Fig. 1a, the average number of assigned controllers in the second scenario for Sprint at $m = 1$ and $m = 2$ is the same. In addition, the

TABLE III: Load distributions on the controllers.

	Resilience	Sprint	ATT NA (1)	PSINet	ATT NA (2)	UUNET
Scenario 1	m=0	(3, 5, 3.66, 1.15)	(5, 7, 6, 1.41)	(4, 10, 8, 3.56)	(7, 10, 8.33, 1.52)	(5, 10, 8.40, 2.30)
	m=1	(5, 10, 7.33, 2.51)	(7, 10, 8, 1.73)	(8, 10, 9.60, 0.89)	(10, 10, 10, 0)	(6, 10, 9.22, 1.30)
	m=2	(5, 10, 8.25, 2.36)	(6, 10, 9, 2)	(4, 10, 9, 2.13)	(8, 10, 9.37, 0.74)	(7, 10, 9.69, 0.85)
Scenario 2	m=0	(3, 8, 5.50, 3.53)	(2, 10, 6, 5.65)	(4, 11, 8, 3.60)	(7, 10, 8.33, 1.52)	(5, 13, 10.50, 3.78)
	m=1	(11, 11, 11, 0)	(5, 10, 8, 2.64)	(6, 13, 9.60, 2.70)	(8, 12, 10, 2)	(6, 14, 10.50, 2.67)
	m=2	(11, 11, 11, 0)	(3, 9, 7.20, 2.49)	(4, 13, 9, 2.56)	(8, 12, 9.37, 1.92)	(8, 14, 11.45, 2.11)
Scenario 3	m=0	(3, 8, 5.50, 3.53)	(2, 7, 4, 2.64)	(4, 10, 8, 3.46)	(3, 9, 6.25, 2.50)	(3, 12, 7, 3.34)
	m=1	(3, 10, 7.33, 3.78)	(4, 12, 8, 4)	(4, 16, 12, 5.65)	(3, 14, 8.33, 3.72)	(4, 16, 9.33, 3.74)
	m=2	(4, 11, 6.60, 2.96)	(5, 12, 9, 2.94)	(7, 18, 14.40, 4.27)	(3, 14, 9.37, 3.50)	(4, 15, 9.69, 3.72)

difference between the maximum propagation latencies in the mentioned resilience levels is negligible (Fig. 2c). Thus, the second scenario is promising for the Sprint topology.

The third scenario (randomly chosen capacities and demands for the controllers and switches, respectively) shows the worst behavior with regard to load imbalance for all of the topologies, which is more obvious for the topologies with larger network size. Moreover, ATT NA (2) has a better load imbalance compared with its older version as well as other topologies for almost all of the three scenarios. One of the main reasons is better connectivity and path redundancy in the topology graph (it has higher average node degree compared with other topologies).

V. Conclusion

In this paper, we proposed a new formulation for the resilient controller placement problem, which takes into account the capacity of the controllers as well as the demands of the switches. Minimizing the total incurred cost (including the cost of deployment, the propagation latency, and the number of required controllers) achieved while considering different resilience levels to enhance the resilience of the controller plane. The obtained results on the real tier-1 network topologies show the effectiveness of the proposed solution for improving the resilience of the controllers while satisfying other performance metrics (e.g., latency). We believe that such a formulation and analysis would be beneficial for the network operators not only during the initial design, but also during the incremental design (as is the case with ATT NA) of their SDN-based networks. Future research directions involve designing heuristic/approximation algorithms to deal with large network sizes as well as testing more random/synthetic topologies to gain useful insights on the results.

Acknowledgment

This work is supported in part by NSERC, CFI and BCKDF.

References

[1] B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[2] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *Communications Surveys & Tutorials, IEEE*, vol. PP, no. 99, pp. 1–20, 2013.

[3] "Software-Defined Networking: The New Norm for Networks," *ONF White Paper*, 2012.

[4] A. S. da Silva, P. Smith, A. Mauthe, and A. Schaeffer-Filho, "Resilience Support in Software-defined Networking: A Survey," *Computer Networks*, vol. 92, pp. 189–207, 2015.

[5] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 473–478, Sep. 2012.

[6] G. Yao, J. Bi, Y. Li, and L. Guo, "On the Capacitated Controller Placement Problem in Software Defined Networks," *Communications Letters, IEEE*, vol. 18, no. 8, pp. 1339–1342, 2014.

[7] A. Sallahi and M. St-Hilaire, "Optimal Model for the Controller Placement Problem in Software Defined Networks," *Communications Letters, IEEE*, vol. 19, no. 1, pp. 30–33, 2015.

[8] L. Yao, P. Hong, W. Zhang, J. Li, and D. Ni, "Controller Placement and Flow-based Dynamic Management Problem towards SDN," in *IEEE ICC Workshop (ICCW)*, 2015, pp. 363–368.

[9] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive Resource Management and Control in Software Defined Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 18–33, 2015.

[10] Y. Zhang, N. Beheshti, and M. Tatipamula, "On Resilience of Split-Architecture Networks," in *IEEE GLOBECOM*, 2011, pp. 1–6.

[11] N. Beheshti and Y. Zhang, "Fast Failover for Control Traffic in Software-defined Networks," in *IEEE GLOBECOM*, 2012, pp. 2665–2670.

[12] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware Controller Placement for Software-Defined Networks," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013, pp. 672–675.

[13] M. Guo and P. Bhattacharya, "Controller Placement for Improving Resilience of Software-Defined Networks," in *Networking and Distributed Computing (ICNDC)*, 2013, pp. 23–27.

[14] L. F. Muller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: An Enhanced Controller Placement Strategy for Improving SDN Survivability," in *IEEE GLOBECOM*, 2014, pp. 1909–1915.

[15] F. J. Ros and P. M. Ruiz, "On Reliable Controller Placements in Software-Defined Networks," *Computer Communications*, vol. 77, pp. 41 –51, 2016.

[16] F. J. Ros and P. M. Ruiz, "Five Nines of Southbound Reliability in Software-defined Networks," in *Proceedings of the Third ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, 2014, pp. 31–36.

[17] M. S. Lawrence V. Snyder, Daskin, "Reliability Models for Facility Location: The Expected Failure Cost Case," *Transportation Science*, vol. 39, no. 3, pp. 400–416, 2005.

[18] R. Yu, "The Capacitated Reliable Fixed-charge Location Problem: Model and Algorithm," Master's thesis, Lehigh University, 2015.

[19] "GUROBI Optimizer," http://www.gurobi.com/, accessed: 2016-03-25.

[20] M. Winther, "Tier-1 ISPs: What They Are and Why They Are Important," https://goo.gl/GgcTfo, 2006, accessed: 2016-03-27.

[21] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[22] P. Gill, N. Jain, and N. Nagappan, "Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 350–361.

[23] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California Fault Lines: Understanding the Causes and Impact of Network Failures," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 315–326.

[24] Maps of World, "USA Population Map," http://goo.gl/Vqqw0e, accessed: 2014-04-05.

[25] P. M. Eittenberger, M. Großmann, and U. R. Krieger, "Doubtless in Seattle: Exploring the Internet Delay Space," in *Next Generation Internet (NGI)*, 2012, pp. 149–155.