A Resilient Mechanism for Multi-Controller Failure in Hybrid SDN-based Networks

Luis Guillen[§], Satoru Izumi[†], Toru Abe[‡] and Takuo Suganuma[‡]
[§]*Research Institute of Electrical Communication, Tohoku University, Sendai, Japan*
[†]*National Institute of Technology, Sendai College, Sendai, Japan*
[‡] *Graduate School of Information Sciences, Tohoku University, Sendai, Japan*
[‡] *Cyberscience Center, Tohoku University, Sendai, Japan*
Email: lguillen@tohoku.ac.jp, izumi@sendai-nct.ac.jp,{beto, suganuma}@tohoku.ac.jp

***Abstract*—SDN is an emerging network paradigm whose main characteristic is the separation of the Control from the Data Plane, allowing the implementation of innovative, robust, and flexible ways to program networks. Moreover, SDN has recently extended its coverage to Hybrid wired and/or wireless environments. However, SDN-enabled devices must be connected to a controller (i.e., a central entity) to be programmable. Therefore, if multiple controllers fail in a short period, regardless of the ample resource availability in Hybrid SDN environments, the whole network infrastructure, and the overall service are compromised. This article presents a mechanism capable of handling multi-controller failure in SDN by considering the hybrid nature of the infrastructure and the scale of the failure. Preliminary results show that by applying the proposed mechanism, it is possible to increase the device's controller coverage by up to 70% even if half of the controllers are unavailable compared to conventional approaches.**

***Index Terms*—SDN, Network Survivability, Hybrid Networks**

I. Introduction

Software-Defined Networking (SDN) [1] is changing the way networking resources are managed, allowing network programmability by flexibly adapting the decoupled control (C-Plane) from the data plane (D-Plane). Furthermore, although in the early years SDN was implemented almost exclusively for wired Data-center or campus infrastructures [2], recently there have been significant efforts in extending the SDN domain to Hybrid Wired and/or Wireless Technologies, at device [3] and end-host level [4], which will benefit the adoption of Software-Defined Everything (SDX). However, SDN-based infrastructures rely on the connection to a controller (i.e., centralized manager) to decide how to handle the resources. Thus, if the controller is unavailable due to unforeseen reasons (e.g., failure or attack), the device has very limited or no decision-making capabilities, rendering the controller a single-point-of-failure [5].

Therefore, it is of paramount importance to have a fail-over mechanism preventing disconnection to the controller. In early stages fail-over mechanisms were straightforward. In these mechanisms, a device might use an alternative pre-defined connection when the main controller fails [6] or use alternative paths to the controller when part of the infrastructure is suddenly disconnected [7]. However, when a considerable portion of the network is disrupted in a short period, including controllers, these mechanisms are not effective.

To illustrate the issues, consider the following scenario. In a large-scale Hybrid SDN-based network, there can be various controllers in charge of their respective domains. Also, in this environment, some devices are multihomed (i.e., support heterogeneous wired/wireless data plane), and therefore, some segments of the network are wired while others are wireless. In an ideal state, all elements are functioning correctly and connected to a controller; so that the data transmission can be easily handled. Assume the controllers have been strategically placed so that they cover a considerable number of devices. Moreover, alternative paths were set in advance so that, if the controller connection (or the controller itself) fails, there will be at least an alternative path from the device to a controller. However, imagine a large earthquake suddenly hits a critical area in the network leading to a progressive failure of the devices. In this case, all the proactive measures that could have been implemented are rendered unusable since various controllers or routes that are not available anymore. Even if the devices might have some reactive measures for re-gaining connectivity, such as calculating a new path to a particular controller. When these measures are to be installed, some of the devices/paths/controllers can be unavailable (or unreachable). Therefore, although there might be various functioning devices that survived the disruption, when they lose connection to a controller, they also lose all programmability. We call these devices *non-operational* as they cannot be used, despite being functional and having various capabilities.

In a prior work [8], we addressed C-Plane reliability on SDN for large-scale disasters; however, we did not consider heterogeneous network environments. Therefore, this article extends our prior work and presents a multi-controller failure resilient mechanism that considers the characteristics of Hybrid SDN-based networks for improving the network coverage of non-operational devices. Preliminary results show up to a 70% increase in coverage and up to 80% improvement in the transmission success rate compared to conventional reactive and proactive methods.

The remainder of this article is organized as follows, Section II briefly discusses the related work. Then, Section III presents the proposed mechanism, which is evaluated in Section IV. Finally, Section V concludes this paper with some final thoughts and future work.

II. Related Work

To the best of our knowledge, multi-controller failure in SDN-based Hybrid networks has not been fully explored. However, various authors proposed C-Plane failure recovery mechanisms assuming that the D-Plane was homogeneous and there was a single-element failure [9]–[17]. Most of these approaches add more overhead or require a modification all the devices [18], which is not practical on a large scale. The extra overhead will result in a long converge time and the excessive (unnecessary) computation; and due to the heterogeneity of the devices, it would be very difficult to modify them all to support new features.

Other authors also proposed C-Plane resiliency by design [19]–[23], so that, the controller placement and alternative paths are pre-calculated. However, as described in Section I, most of these methods cannot cope with a large-scale failure of multi-controllers in a short period. Since, by the time they re-calculate a new alternative, the network configuration would have been already gravely impaired by the failure.

It is also worth mentioning the pioneering work of Never-Die Network (NDN) [3], [24], [25], which is a conceptual architecture to enable communication services disrupted by disasters using heterogeneous D-Planes for Network resilience (e.g., mobile and aerial). However, in NDN, the C-Plane resilience was not addressed as they only considered a single SDN domain.

This study complements the existing work by presenting a mechanism to enhance C-Plane's resiliency in Hybrid SDN-based networks when a large portion of the network is disrupted due to a large-scale failure. However, contrary to some of the related work, the proposed mechanism does not rely on device/protocol modification or pre-defined controller's placement, or pre-calculated alternative paths.

III. Proposed Resilient Mechanism

The mechanism aims to enhance the network survivability by extending the C-Plane coverage in SDN-enabled devices when a large-scale failure disrupts the network. This section summarizes the proposal.

A. Network Model

Before delving into the proposal, this sub-section presents the assumed network model. Fig. 1 depicts the model divided into two layers. At the bottom layer, the D-Plane is represented by a graph $G\left(V, E\right)$ of a set of nodes V, and edges E, whose parameters are summarized in Table I. We assume that the position of a node $(x_i,\ y_i)$ is known, and although some devices support multihoming, there will be a single active edge $(e_{i,j})$ between two nodes, which is from a specific type of connection (i.e., wired or wireless). Note that the cost of the path between two nodes $(PC_{s,t})$ depends on the edge's cost $(\varphi_{i,j})$, which is a parameter that varies according to the stage in the mechanism.

At the C-plane, the graph $G'\left(C, D_m, Z_m\right)$ where the set of controllers C is defined as follows: if a node $v_1 \in V$ is physically connected to a controller (i.e., they are in the same network facility) then $\exists\ c_1$; for instance, in Fig. 1 since there are controllers attached to v_1 and v_6 then the set of controllers is defined as $C = \{c_1\,, c_6\}$.

TABLE I
Basic Parameters in the Network Model.

Parameter	**Description**
v_i	Network devices
$x_i,\ y_i$	Node position
$e_{i,j} = e_{j,i}$	Edge between v_i and v_j
$b_{i,j}$	Edge bandwidth
$\varphi_{i,j}$	Edge cost
$p_{s,t}$	Path from s to $t \in V$
$PC_{s,t}$	Path cost $\sum_{\forall e_{i,j} \in p_{s,t}} \varphi_{i,j}$

The Domain of c_m is the set D_m, which is comprised by all the nodes v_i associated to the controller c_m. For instance, the domain D_1 of the controller c_1 in Fig. 1 is defined as $D_1 = \{v_1\,, v_2\,, v_4\}$.

The connection from the controller to devices and vice-versa, the set $Z = O_m \cup I_m$ represents out-of-band (i.e., direct) and in-band (i.e., indirect shared with D-Plane) connections for each controller c_m to each node in its domain D_m. For simplicity, we assume that there is only one out-of-band controller connection. However, there might be a number (k) of in-band connections, which is, in principle, k different paths with minimum cost from c_m to $v_i \in D_m$.

Finally, the inter-domain communication (i.e., connection from c_1 to c_6 in Fig. 1) uses the path $p_{i,j}$ from $v_i(c_i)$ to $v_j(c_j)$ such that $PC_{i,j}$ is minimum.

B. Objective Function

Assume that F is the set of failed controllers within an affected area A, such that $F \subseteq C \neq \emptyset$. The devices $v_i \in V$

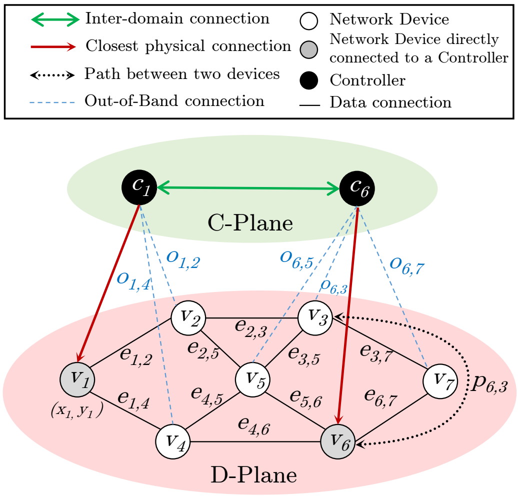

Fig. 1. C- and D-Plane Network Model.

associated in the domain D_i of a controller $c_i \in F$ lose connectivity at the C-Plane. At the D-Plane, the network devices and the associated links $e_{i,j}$ within the affected area A will also fail, consequently failing the on-going data-flows ($P_{s,t}$) as well. Let $F_{s,t} \subseteq P_{s,t}$ be the set of failed flows; since there are devices $v^*{}_i \in D_i$ which are not affected by A or might be non-operational, the operation of flows $F^*{}_{s,t} = P_{s,t} \cap F_{s,t}$ should continue as long as there is an available route through the operational nodes and links. However, the devices $v^*{}_i$ cannot request an updated path to c_i due to connectivity loss to the controller; therefore, those devices need to reconnect to any alternative controller (k) and then recalculate the paths of the failed data-flows.

We formulate the objective function as shown in (1):

$$\min_{i\in V^*}\left(\sum_{\forall i,c_k\in D^*{}_k}\sum_{i,j\in E^*} PC_{i,c_k}x_{ij}^{ic_k} + \sum_{\forall s,t\in F_{s,t}}\sum_{i,j\in E^*} PC_{s,t}y_{ij}^{st} + \sum_{\forall s,t\in F^*{}_{s,t}}\sum_{i,j\in E^*} PC_{s,t}z_{ij}^{st}\right) \tag{1}$$

$$\textbf{s.t.}\begin{cases} x_{ij}^{ic_k}, y_{ij}^{st}, z_{ij}^{st} & \in \{0,1\} \\ x_{ij}^{ic_k}, y_{ij}^{st}, z_{ij}^{st} & = \begin{cases} 1 & \text{if } x_{ic_k}, y_{st}, z_{st} \text{ traverses } e_{i,j} \\ 0 & \text{otherwise} \end{cases} \\ x_{ic_k}, y_{st}, z_{st} & \text{are the safest path} \end{cases}$$

$PC_{i,c_k}x_{ij}^{ic_k}$ refers to the C-Plane, and the objective is to maintain the connectivity to the closest $k \geq 1$ controllers, such that the cost of establishing those new connections using the safest paths in the surviving edges E^* should be minimized. The second and third terms refer to the cost to maintain the connectivity of the D-Plane failed data-flows $F_{s,t}$ (in the first case) and the surviving flows that might fail in a near-future $F^*{}_{s,t}$ (in the second case) by redirecting the flows through the safest paths. Note that the *safest path* will be determined by the proposed mechanism depending on various factors (detailed in the following section), such as the distance to the epicenter in A and the type of technology used in the underlying infrastructure.

C. Three-stage Resilient Mechanism

In short, the proposal considers a failure warning time (Δt), which triggers a three-stage mechanism to protect the connectivity in both planes, taking into account the risk and performance in its heterogeneous deployment. Note that Δt might vary depending on the type of assumed failure. For instance, in an earthquake, the failure alert systems are triggered few seconds after the first shock. However, in other cases, this value might require to be as little as few milliseconds.

The overall timeline is a sequential process that spans the whole failure from the alert until the service recovery. The primary goal of the proposed mechanism is to avoid controller disconnection; thus, a pre-determined number (k) of alternative paths to controllers is calculated around the disaster perimeter to the k-nearest controllers using Region-Disjoint and Maximum Disjoint Paths (adapted Floyd-Warshall algorithm). The criteria to assess the safest paths with the minimum weight is based on the *Risk Factor Index* (RFI), which varies according to the stage. In the first phase (Controller Disconnection Avoidance CDA), the RFI of a node is defined as in (2).

TABLE II
CONNECTION RISK AND PERFORMANCE ANALYSIS.

Technology	Reliability	Performance
Ethernet	Low	Medium
Fiber	Low	High
Wireless	Medium	Medium
Satellite	High	Low

$$RFI_i = \frac{r_{max}}{||d(v_i, \epsilon)||} * \alpha \tag{2}$$

Where

- $d(v_i, \epsilon)$ is the Euclidean distance from the node v_i to the epicenter of the disaster (ϵ) with a maximum failure range of r_{max}. Therefore, the closer to the epicenter, the higher the risk of failure.
- α is the risk of the infrastructure where the node is hosted. For instance, since core devices in terrestrial stations have a higher risk of being affected by a disaster $\alpha = high$, while Satellite would have less risk of being affected by a disaster, therefore $\alpha = low$.

Then, in the second stage, *Data Communication Protection* (DCP), once the CDA stage finishes, there will be at least one alternative connection to the controller if the device is reachable. DCP protects the Data Communication by calculating the RFI of the edges as in (3).

$$RFI_{e_{i,j}} = \begin{cases} \frac{r_{max}}{||d(e_{i,j},\epsilon)||} * \beta + \gamma & \text{if } \frac{T_{e_{i,j}}}{T_{max}} \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

Where

- $d(e_{i,j}, \epsilon)$ is the Euclidean distance from the edge to the epicenter of the disaster (ϵ).
- β is the risk associated with the edge, based on the characteristics of the connection type described in Table II. Note that we only use the reliability parameter at this stage. Moreover, based on preliminary experiments, we found out that the values of α and β have no influence in the calculation as long as we used the same characterization for the level of risk (i.e., the same values for low, medium, and high for both α and β).
- γ is an adjustment factor in ensuring the disjoint property for each path. For example, in the CDA phase, we use the number of in-band connections going through a link.

- $T_{e_{i,j}}$ is the expected time of failure of $e_{i,j}$ after the failure alert at Δt, and T_{max} estimate time of the failure will reach its maximum area A. If that value has been reached, then the link has failed, and therefore the link is unusable.

Finally, the third and last stage, *Disaster Impact Monitoring* (DIM), monitors the impact of the disaster by periodically updating the state of the network at a given update interval T_{update} with a pre-defined maximum number of intervals $n > 1$. The RFI is calculated at time t_n, as in the previous stages with some minor differences. If the node cannot be accessed, it is removed from set V as well as all the associated edges. Note γ at this stage will use the number of data transmissions going throughout the link.

IV. Evaluation

A. Simulator

Current emulation and simulation environments have well-known limitations in terms of C-Plane, especially in terms of distributed and hybrid SDN. For instance, there is no support for multiple in-band connections, domain transition from a failed controller to another (i.e., inter-domain hand-off), or device discovery in hybrid environments using different technologies. Therefore, we developed a custom-made Java simulation that numerically implements the functions described in Section III. However, to simplify the modeling and implementation, we made the following assumptions and abstractions:

- Seamless controller-line handover. If a controller is listed in the devices' database, it will be used immediately. In a real setup it takes at least one RTT to set the connection.
- Seamless inter-domain hand-off. In real SDN-based environments, the East-West bound protocols are still an open issue; unless the controllers have a distributed synchronization mechanism, it takes a long time for a controller to connect to new devices and collect their statistics. However, we assume this is immediate.
- All control traffic is assumed to be correctly functioning while there is at least a controller connection.
- Rules are instantly installed so that the calculated paths can be implemented and available in each device as soon as the calculation finishes.
- The integration of hybrid network protocols is seamless. Therefore, there will be no issues when transitioning from a type of network to another.
- Instant failure detection, thus if a device/link fails is is automatically removed from the graph and the other devices database. Note that this might take a considerable amount of time in real networks.

It is also worth mentioning that, since the objective of this paper is to show the potential of the proposed mechanism rather than an actual network deployment, we only implemented the basic required network functionality. For instance, we used elementary data structures for nodes, edges, paths, and controllers; each containing the primary elements to perform the graph operations described in the previous sections. Moreover, we simply subtract a specific amount from the data transfer depending on the set connection speed; therefore, advanced features such as congestion control or others present in real networks are not implemented.

TABLE III
Experimental Parameters.

Parameter	Value
Topology	Gabriel graphs
Maximum Bandwidth ($B_{i,j}$)	100 Mbps (Ethernet), 1000 Mbps (Fiber), 600 Mbps (Wireless), 40 Mbps (Satellite)
Test Surface	30 × 20 km^2
Node Risk (α)	1 (Low), 2 (medium), 3 (high)
Edge Risk (β)	1 (Low), 2 (medium), 3 (high)
Number of Nodes	200
Number of Flows (transmissions)	25 per test
Amount of data per Flows	250 to 1000 MB
Number of Controllers	10
Failure alert (Δt)	10 s
Checkpoint time (T_{update})	30 s
Maximum affected range (r_{max})	5 km
Disaster propagation speed (s)	50 m/s
Number of Alternative Controllers (k)	2

Concerning the *failure model*, we assume a simplistic model consisting of a circular-shaped disaster with a random epicenter ϵ, and a foreseen maximum damage radio r_{max}. The disaster area expands from ϵ homogeneously at speed s so that a device v_i or the links attached to the device will fail if the Euclidean distance from the node or link is within the affected area. Table III summarizes the values of the parameters.

All the experiments were conducted using a single virtual machine running Ubuntu 16.04 LTS, with six CPUs Intel Xeon(R) E5-2650 v4 @ 2.20 GHz and 16 GB of memory.

B. Experiment Procedure

For every run of the experiment, a random topology is created using synthetic Gabriel graphs to build the network topology. We used Gabriel graphs since it creates the closest structure geographic models compared with physical networks [26]. The proportion of created nodes and edges uses a negatively skewed distribution as follows: For nodes, Terrestrial stations (80%), Mobile Stations (15%), and Satellite Stations (5%). For edges, Fiber (70%), Wifi (15%), Ethernet (10%), and Satellite (5%) lines, all of them using their respective bandwidths (as described in Table III).

Once all nodes and edges are created, each node is assigned to a single controller, randomly placed in the topology.

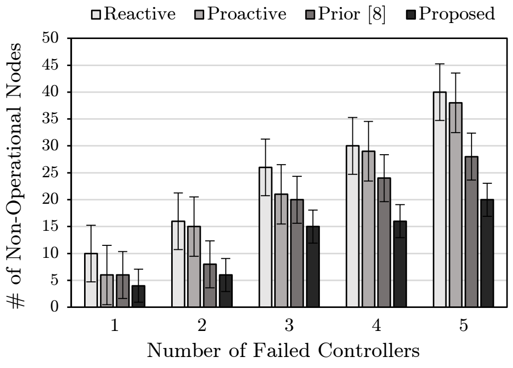

Fig. 2. Number of Non-Operational Nodes per Approach.

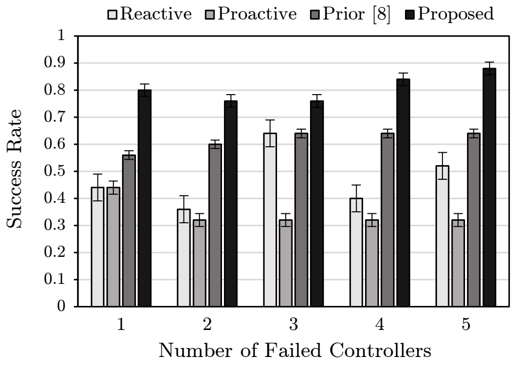

Fig. 3. Success Rate of Data Transmissions per Approach.

Then, at $t = \Delta t$, we set a randomly available bandwidth value ($b_{i,j}$) to each link according to its type. Moreover, 25 transmissions of various sizes (from 250 to 1000 MB) initiate simultaneously. Finally, when the disaster damage starts to spread from a random position, the status is updated every given interval until all the transmission has finished (either successfully or failed). Note that we used the same seeds to create the pseudo-random parameters throughout each simulation so that all test approaches go over the same characteristics.

C. Comparison Approaches and Metrics

To evaluate the proposal's effectiveness, we compare:

1) **Reactive:** This approach calculates the k alternative controllers and their paths immediately after a failure is detected, using the shortest paths to those controllers.
2) **Proactive:** In this approach, k alternative controllers and their corresponding shortest paths are pre-installed in all devices. If all these controllers fail to connect, then a reactive-like approach is adopted.
3) **Prior:** This approach is a prior work [8], which does not consider heterogeneous hybrid SDN environments for the RFI calculation.
4) **Proposed:** This approach is the proposed mechanism, which implements the functions described in Section III.

Moreover, the metrics measured were as follows:

- Percentage of non-operational nodes, which refers to the number of network devices that did not fail, but not connected to any controller.
- Success rate, which refers to the number of successful transmissions at the D-Plane over all the initiated flows.
- Number of path re-routes, meaning the number of times a D-Plane transmission needed to be re-calculated as the previous one(s) failed.

D. Results

The experiment was conducted hundreds of thousands of times, but we only considered the results when at least one of the controllers failed.

1) Non-operational nodes: Fig. 2 depicts the results obtained of non-operational nodes. As observed, the proposed approach increased the device's controller coverage by up to 70% non-operational nodes than the other approaches. Note that the difference in coverage is more evident as more controllers fail. For instance, when 5 of the controllers failed, the proposed mechanism could reduce the number of non-operational devices by half compared to the reactive approach. These results show that it is possible to improve the devices' controller coverage.

2) Success rate: Fig. 3 summarizes the results obtained. As observed, from the 25 transmissions, on average, about 20 transmissions (80%) finished successfully regardless of the number of failed controllers, compared to 9 (38%), 8 (34%), and 15 (61%) using the reactive, proactive, and the prior approach. Note that the success rate of data transmissions was very high considering the number of failed devices due to the safest paths. It is also worth mentioning that the propagation speed used in the experiments was very conservative (50 m/s); in scenarios where this speed was some scales faster, only our prior and proposed approach can finish a substantial percentage of transmissions, as we showed in [8].

3) Path re-routes: As observed in Fig. 4, due to link/node failure used for the data transmission (which also carries the in-band connection to the controllers), it was necessary for

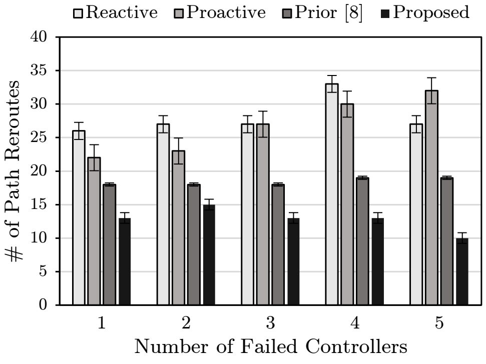

Fig. 4. Percentage of Data Reroutes per Approach.

each approach to re-calculate an alternative path. Note that the proposed method considerably reduced the number of reroutes, especially compared with the reactive and proactive strategies.

From these results, we can conclude that the proposed mechanism has great potential to increase the controller coverage of devices, increasing the C- and D-Plane reliability. However, these values might vary as we include other realistic parameters. Nevertheless, this study aimed to show the influence of SDN hybrid environments in service provision when a multi-controller failure occurs. Therefore, we leave as future work the inclusion of other realistic parameters.

V. Conclusions and Future Work

Despite having extended its coverage to Hybrid wired/wireless, SDN-enabled devices must connect to the Controller to allow flexible resource management. This article presented a mechanism to improve the resilience in SDN Networks when multiple controllers fail in a short period in these hybrid environments.

The proposal uses a failure alert to trigger a three-stage process. Preliminary results show a higher coverage of devices at the C-Plane and a higher completion rate at the D-Plane, which significantly benefit the overall survivability and QoS.

For future work, we are considering using mobile/aerial stations to re-connect the remaining non-operational devices, as in the current state of the proposal, we do not deploy any self-healing mechanisms for devices, as done by some other authors [9], [10]. This would also help to solve the so-called *Sub-network isolation* [12] or *SDN Domain-Splitting* [9], which refers to a group of devices isolated by the impact of a failure. Therefore, it would be interesting to address this feature. It is also necessary to further evaluate some of the most critical parameters, such as the number of alternative controllers (k) or the minimum failure alert time (Δt). Finally, we plan to incorporate more realistic elements in the simulation and implement them using more standard tools.

References

[1] Open Networking Foundation (ONF), "Software-Defined Networking (SDN) Definition," Available online: https://www.opennetworking.org/sdn-definition/ (accessed Jan 2021).

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," SIGCOMM Comput. Commun., Vol. 38, no. 2, pp. 69-74, 2008.

[3] Y. Shibata, N. Uchida, and N. Shiratori, "A Resilient Network and Information Management System for Large Scale Disaster," in *30th Int. Conf. on Advanced Information Networking and Applications Workshops*, Crans-Montana, Switzerland, 23–25 March 2016, pp. 655–660.

[4] I. Martinez-Yelmo, J. Alvarez-Horcajo, J. A. Carral, and D. Lopez-Pajares, "eHDDP: Enhanced Hybrid Domain Discovery Protocol for network topologies with both wired/wireless and SDN/non-SDN devices Computer Networks,"in *Computer Networks*, vol. 191, pp.1–16, 2021.

[5] L. Sidki, Y. Ben-Shimol, and A. Sadovski, "Fault tolerant mechanisms for SDN controllers," in *Proc. IEEE Conf. on Network Function Virtualization and Software Defined Networks*, Palo Alto, CA, 7–10 November 2016, pp. 173–178.

[6] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Fast failure recovery for in-band OpenFlow networks," in *Proc. 9th Int. Conf. on the Design of Reliable Communication Networks*, Budapest, Hungary, 4–7 March 2013, pp. 52–59.

[7] Y. Hu, W. Wendong, G. Xiangyang, C. H. Liu, X. Que, S. Cheng, "Control traffic protection in software-defined networks," in *Proc. 2014 IEEE Global Communications Conference*, Austin, TX, 2014, 8–12 December 2014, pp. 1878–1883.

[8] L. Guillen, H. Takahira, S. Izumi, T. Abe, and T. Suganuma, "On Designing a Resilient SDN C/M-Plane for Multi-Controller Failure in Disaster Situations," in *IEEE Access*, vol. 8, pp. 141719–141732, 2020.

[9] T. Omizo, T. Watanabe, T. Akiyama, K. Iida, "ResilientFlow: Deployments of Distributed Control Channel Maintenance Modules to Recover SDN from Unexpected Failures," *IEICE Trans. Commun.*, Vol. E99-B, No. 5, pp. 1041–1053, 2016.

[10] M. Osman, J. Núñez-Martínez, and J. Mangues-Bafalluy, "Hybrid SDN: Evaluation of the impact of an unreliable control channel," in *Proc. 2017 IEEE Conf. on Network Function Virtualization and Software Defined Networks*, Berlin, Germany, 6–8 November 2017, pp. 242–246.

[11] S. A. Astaneh, and S.S. Heydari, "Optimization of SDN Flow Operations in Multi-Failure Restoration Scenarios," in *IEEE Transactions on Network and Service Management*, Vol. 13, No. 3, pp. 421–432, 2016.

[12] T. Hirayama, M. Jibiki, and H. Harai, "Designing Distributed SDN C-Plane Considering Large-Scale Disruption and Restoration," in *IEICE Transactions on Communications*, Vol. 102, No. 3, pp. 452–463, 2019.

[13] A.S.M. Asadujjaman, E. Rojas, M.S. Alam, and S. Majumdar, "Fast Control Channel Recovery for Resilient In-band OpenFlow Networks," in *Proc. 2018 4th IEEE Conf. on Network Softwarization and Workshops*, Montreal, Canada, 25–29 June 2018, pp. 19–27.

[14] B. Görkemli, S. Tatlıcıoğlu, A.M. Tekalp, S. Civanlar, and E. Lokman, "Dynamic Control Plane for SDN at Scale," in *IEEE Journal on Selected Areas in Communications*, Vol. 36, No. 12, pp. 2688–2701, 2018.

[15] K. Chan, C. Chen, Y. Chen, Y. Tsai, S. W. Lee, and C. Wu, "Fast Failure Recovery for In-Band Controlled Multi-Controller OpenFlow Networks," in *Proc. Int. Conf. on Inf. and Comm. Technology Convergence*, Jeju, South Korea, 17–19 Oct 2018, pp. 396–401.

[16] D. Lopez-Pajares, J. Alvarez-Horcajo, E. Rojas, A. S. M. Asadujjaman, I. Martinez-Yelmo , "Amaru: Plug&Play Resilient In-Band Control for SDN," in *IEEE Access*, Vol. 7, pp 123202–123218, 2019.

[17] M. Silva Freitas, R. Oliveira, D. Molinos, J. Melo, P. Frosi Rosa, and F. de Oliveira Silva, "ConForm: In-band Control Plane Formation Protocol to SDN-Based Networks," in *Proc. Int. Conf. on Information Networking*, Barcelona, Spain, 7–10 Jan 2020, pp. 574–579.

[18] Y. Zhang, J. Tourrilhes, Z.L. Zhang, and P. Sharma,"Improving SD-WAN Resilience: From Vertical Handoff to WAN-Aware MPTCP," in *IEEE Tran. on Network and Service Management*, vol. 18, no. 1, pp.347–361, 2021.

[19] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. 25th International Teletraffic Congress*, Shanghai, China, 10–12 Sept 2013, pp. 1–9.

[20] J. Lu, Z. Zhang, T. Hu, P. Yi and J. Lan, "A Survey of Controller Placement Problem in Software-Defined Networking," in *IEEE Access*, vol. 7, pp. 24290–24307, 2019.

[21] L. Müller, R. Oliveira, M. Luizelli, L. Gaspary, and M. Barcellos, "Survivor: An enhanced controller placement strategy for improving SDN survivability," in *Proc. IEEE Global Communications Conference*, Austin, TX, USA, 8–12 December 2014, pp. 1909–1915.

[22] N. Beheshti, and Y. Zhang, "Fast failover for control traffic in Software-defined Networks," in *Proc. IEEE Global Communications Conference*, Anaheim, CA, 3–7 December 2012, pp. 2665–2670.

[23] S. S. Savas, M. Tornatore, M. F. Habib, P. Chowdhury, and B. Mukherjee, "Disaster-resilient control plane design and mapping in software-defined networks," in *Proc. IEEE 16th Int. Conf. on High Performance Switching and Routing*, Budapest, Hungary, 1–4 July 2015, pp. 1–6.

[24] T. Suganuma, G. Kitagata, T. Kato, and N. Shiratori, "Structure of Never Die Network Service Organization in Wireless Networks," in *Proc. 2003 IEICE Gen. Conference*, Niigata, Japan, pp. 376, 2003. (Japanese)

[25] N. Shiratori, N. Uchida, Y. Shibata, and S. Izumi, "Never Die Network towards Disaster-resistant Information Communication Systems," in *ASEAN Engineering Journal*, vol. 1, no. 2, pp.1–22, 2013.

[26] E.K. Çetinkaya, M.J.F. Alenazi, Y. Cheng, A.M. Peck, and J.P.G. Sterbenz,"On the fitness of geographic graph generators for modelling physical level topologies," in *Proc. 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Almaty, Kazakhstan, 10–13 Sept 2013, pp. 38–45.