Migrating From Legacy to Software Defined Networks: A Network Reliability Perspective

Yaser Al Mtawa, Anwar Haque, and Hanan Lutfiyya

***Abstract*—Designing survivable communication networks to achieve carrier-grade five-nines reliability is of paramount importance for the network operators. This article addresses service reliability and its related aspects such as nodal reachability, network connectivity, and edge-disjoint routing in both traditional networks and software defined networks (SDNs). The proposed roadmap is based on two phases: Fundamental analytical phase and performance evaluation phase. In the first phase, a graph operator is defined to analyze the characteristics of the reliability metric and its associated reachability feature. This phase will focus on both the macro- and micro-level properties of reliability. In the second phase, we exploit the analysis in the former phase to get an insight into the performance evaluation of traditional and SDN-based networks against the reliability metric, and then calculate the statistical significance of the mean difference of their reliability values. Reliability under edge-disjoint paths to avoid resource competition is also investigated. Various types of topologies are utilized to test the service reliability of both architecture designs. Extensive simulation results show that SDN-based networks have comparable performance to its legacy counterpart against the operational reliability metric. Our findings not only shed light on enhancing reliability using edge-disjoint paths under link failure scenarios but also expected to benefit the operators to achieve their service level objectives while migrating from legacy to SDN-based platform.**

***Index Terms*—Legacy network, network reliability, quality of service (QoS), software defined networking (SDN).**

I. Introduction

LEGACY communication networks have both control and data planes combined in a network node. A legacy network supports and embeds several reliability protocols that range from multipath routing to element failure (e.g., detection and handling mechanisms) [1]. However, different types of network communication technologies come with different standards and evolve over time, which makes the core of a legacy network difficult to manage, e.g., a mandate of using low-level vendor-specific commands to update each network device. This complexity of the legacy network hinders the growth of data traffic and the reliability of services [2]–[4].

On the other hand, software-defined networking (SDN) decouples control and data planes [5]. An SDN controller offers a centralized control point that can collect information and determine the best forwarding policies. It can detect and handle element failures, and then provide alternative routes. This property makes SDN an excellent candidate to support network reliability [6]. However, unlike a legacy network, SDN creates a new network plane that connects the decoupled data and control planes, viz. a *control path* network [1]. The multidomain SDN architecture poses further complexity on the network reliability management. These reliability challenges are equally applicable to both legacy and SDN-based systems. The question is now which one is more reliable: Traditional networks or SDN-based networks?

Network reliability measures to what extent the network can stay operational with a minimum level of requirements when facing operational stress [7]. That is, the continuity of service that implements the system function (i.e., correct service) [8]. Reliability is an attribute of system dependability which is the ability to avoid frequent or sever service failures more than acceptable level. Other attributes of dependability include availability (readiness for correct service), robustness (dependability with respect to a specific class of faults ability to tolerate failures), safety, integrity, and maintainability [8]. In communication networks, operational stress could hinder the communication between devices, interrupt the regular network operation, and could cause severe degradation at the service level. There are several causes/faults that can impact network reliability. However, network element faults, transmission, and routing faults cover a wide range of faults. In this article, we consider these three faults: 1) A network element fault: The faults of elements such as devices (i.e., nodes) and links can lead to data transmission obstacles between the various parts of the network such as switches, routers, servers, etc. Several network elements may then be isolated from the main network which will greatly impact the services of the whole system; 2) transmission fault: If there is a data transmission fault between any source and destination nodes in the network, the network will suffer an error in the service state, thus affecting the QoS; 3) Routing fault: A stream of traffic packets will not be able to reach its destination element if a routing fault occurs. The destination node, in this case, will remain waiting for packets that will not arrive at least not before the retransmission. This may cause the destination node to not receive the whole stream in a timely manner, thus, affecting the service delivery. However, there is a correlation between the above faults, as element failure causes transmission

Manuscript received May 28, 2020; revised August 30, 2020 and February 10, 2021; accepted March 8, 2021. Date of publication April 13, 2021; date of current version November 30, 2021. Associate Editor: S.-Y. Hsieh. *(Corresponding author: Yaser Al Mtawa.)*

The authors are with the Department of Computer Science, University of Western Ontario, London, Ontario N6A 3K7, Canada (e-mail: yalmtawa@uwo.ca; ahaque32@uwo.ca; hlutfiyy@uwo.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TR.2021.3066526.

Digital Object Identifier 10.1109/TR.2021.3066526

and routing faults. To avoid possible interference between the faults, we seek to employ the smallest number of faults that can still characterize the network reliability and its related aspects. In this study, we focus on the element faults to address network reliability.

In legacy networks, a router identifies a link failure and establishes an alternative path to push the packets through. On the other hand, in SDN-based networks, a switch lacks intelligence and the global knowledge of network elements' connectivity to establish an alternative path in case of an element/link failure. It relies on the controller to establish such an alternative path. Packets that are supposed to flow through the failed element/link will be dropped until a controller identifies this failed element and updates the entries of forwarding policies in all affected switches. However, in SDN networks, link failure protection is also possible. For example, multitunnel repair paths can be constructed by leveraging multiple SDN switches. Multitunnel repair paths will significantly increase the number of potential alternative paths that can be used upon link failure [9].

In communication networks, reliability is usually analyzed by assessing the host/terminal connectivity [10]. Consequently, reliability analysis includes *2-terminal reliability*, *K-terminal reliability*, and *all-terminal reliability*.

Research works on SDN addressed various topics such as benefits, concepts, challenges, cost savings, ease of management, and provisioning [11]–[14]. Economic analysis of SDN showed the impact of various network failure scenarios on network economics regarding CAPEX (Capital Expenditure), OPEX (Operational Expenditure), and revenue loss [15]. Hybrid SDN networks (i.e., partial deployment of SDN-enabled elements) were also proposed in multiple research work as shown in recent survey articles [16], [17]. These works describe the migration from legacy networks to SDN networks. Research studies also highlighted the characteristics of SDN-based architecture over traditional architecture in terms of programmability, centralized control, network flexibility, efficient configuration, etc. [18]. However, few papers focus on the reliability evaluation of SDN-based networks [19]. The most recent survey paper about performance issues in SDN-based data center [20] showed a lack of specialized reliability-based research regarding SDN-based architectures. A reliability comparison study is important because it highlights the strengths and weaknesses of both networks. This triggers the need for further research in network design to improve service availability provided by a network operator. Real-time services, for example, require stringent packet-delivery measures that mandate a high-reliable network. In our previous paper [21], we showed the importance of SDN in improving the QoS metrics in communication networks. Although SDN-based networks are easier to manage and enforce security and forwarding rules, they are still not fully deployed for many reasons. The large budget required is one of the major reasons [16]. This is because the migration to SDN-based network is not a priority investment for many network operators despite the cost efficiency of SDN-related network in the long run. Another reason is perhaps the lack of specialized research studies that address reliability in SDN-based networks. This motivated us, in this article, to provide a further study to analyze, evaluate, and compare reliability metric for both legacy and SDN-based networks and answer the research question "Does the migration from traditional network to SDN-based network enhance the reliability?" The main contributions of this article are as follows.

1) We study detailed characteristics of reachability and reliability regardless of network type. Link failure is defined as a stochastic-based iterated graph operator that can be applied on any network graph and any failure modes. We show that under link failure, the existence of reachability (i.e., the readiness to provide a service) is not enough for guaranteed network reliability (i.e., the continuity to provide that service). We also show the importance of edge-disjoint paths to enhance network reliability. Based on our analysis of edge-disjoint paths, an efficient polynomial-time algorithm can be designed to calculate the exact K*-pair reliability*.
2) We propose a reliability model using exponential-distribution-based link failure for the sake of assessment of network *all-pair reliability*, which is defined as the probability that the two nodes of any pair of the link set can successfully send/receive data packets with each other.
3) We comprehensively evaluate and compare the two network designs (i.e., traditional versus SDN-based networks) against the reliability metric. Various network types: Tree, mesh, and hybrid are used to generally test the reliability of both legacy and SDN-based network designs.

The remainder of the article is organized as follows. Section II presents related work. In Section III, we describe the network modeling, whereas Section IV describes the reliability metric and its related aspects. It is based on the analyses of both macro-level/structural and micro-level/elements' state of a network. Network topologies and performance evaluation, which includes emulation/simulation, experimental settings, results, and discussion are investigated in Section V. Finally, Section VI concludes this article.

II. Related Work

The most recent survey in SDN [22] highlighted challenges and their respective solutions for reliability in SDN. Reliability research in SDN can be divided into two parts: Data plane and control plan. *Data plane* reliability research proposals can be further classified into fast failure detection and fast recovery. *Fast failure detection* can be enhanced by checking the connectivity among neighboring OpenFlow switches [23]. This approach was confirmed to be feasible and effective as the OpenFlow switch-initiated approach for failure recovery was shown to be faster than the one initiated by the controller due to the simplicity of the message exchange procedure [1]. *Fast recovery* from link failure can be achieved through a local fast reroute algorithm with a minimal update of flow entries in SDN [24].

Control plane reliability research has addressed multiple aspects of the SDN controller such as multiple controllers, controller placement, and flow redirection. Research in the use of *multiple controllers* investigates how multiple controllers can be used to cooperate to enhance reliability, response time, and availability metrics. For example, it was employed in a stochastic model to evaluate the reliability of a multilevel SDN architecture where a group of controllers was attached to each other to efficiently act as a cache system managing local networking and forwarding rules [25] in order to enhance reliability. Similarly, a stochastic activity network-based model can leverage multiple controllers to enhance response time and availability metrics under a possible failure of a controller or a cluster of controllers [26], [27].

Controller placement in a network is utilized to enhance the southbound reliability in SDN [28]. This typically focuses on determining the minimal number of placed controllers to cover all network devices and obtain an efficient controller-to-switch assignment, which were also addressed to enhance the reliability of the control plane [29], [30]. The *flow-based redirection* approach was also used to reduce the maximum response time of the controllers. This mechanism requires installing wildcard rules on switches [31]. However, it enhances the response time up to 80% compared to the case without this mechanism. Unlike previous research work, Santos *et al.* [32] proposed an optimal solution for the controller placement problem considering the network robustness at both parts; *control and data planes*. The authors aimed to minimize the average switch–controller and controller–controller delays against a set of failure states.

In addition, a few research studies that evaluate network reliability have been conducted utilizing different approaches at the *macro-level* of network topology: Minimal-cut sets [33], optimization-based class such as the method proposed in [34], enumeration-based class such as a sum of disjoint products method [35], graph-based class such as binary decision diagram (BDD) [36]–[39], decomposition technique [40], [41], and factoring theorem using polygon-to-chain reduction [42], [43]. *Visual assessment* of network robustness variability was proposed by Manzano *et al*. [44]. The authors introduced a robustness surface concept that utilizes the principal component analysis (PCA) to find the most informative robustness metric under a failure scenario. Reliability analysis was addressed in [45] to determine the impact of virtual machines (VMs) migration on the SDN operational lifetime.

The closest research to ours in terms of the performance evaluation of both legacy and SDN-based networks is described in Nencioni *et al*. [33]. This work describes a Markov-based stochastic method to compare the availability of both network architecture designs under correlated failures. Their stochastic method is based on minimal-cut sets, which is a property of network topology (i.e., network's macro-level). Instead, our approach in this article allows us to consider the failure at both the macro- and micro levels in legacy and SDN-based networks by considering the network topology and instantaneous network state. Furthermore, the paper in [33] is more directed to the study of the availability of network elements which is, in this sense, similar to the reachability rather than the reliability.

The aforementioned reliability proposals and frameworks showed that there is always room for reliability improvement. However, none of these proposals shed light on the impact of legacy–to-SDN migration on network reliability.

Our work is unique as it outlines a framework that allows for a study of operational reliability of legacy-to-SDN migration. We investigate the reliability at both the macro level and operational micro level. At the macro level, we identify the coupling between reachability and reliability metrics. To characterize reachability and reliability, we define a graph operator L. It works under any network lifetime distribution, any network size, and any network topology. It also acts as a link remover of the network under evaluation. The network operation is highly dependent on various parameters such as its lifetime distribution, the network size (i.e., the number of network devices), the corresponding topology, etc. Therefore, recognizing L as a *network operator* and understanding its properties, such as transitivity, enable us to provide effective solutions at the operational micro level that prior approaches fall short of. The micro-level information of a network is necessary to capture the instantaneous states of the network elements. At the micro level, several research questions arise, such as at what lifetime the network can achieve a specific reliability threshold? How many should random failed links be allowed to maintain reliability within a specific threshold? At a specific network age, what links should be proactively maintained to keep reliability within an acceptable level?

The differences between the aforementioned frameworks and our proposal are summarized as follows.

1) The frameworks above lack an insightful understanding of the coupling between reachability and reliability metrics. It has been known in the reliability research community that the existence of reliability of communication between two network devices is about the reachability between them [46]. This is perhaps because of the perception that the reliability evaluation depends only on the structural topology of the network rather than on the characteristics of the whole operational path that was employed to transmit data traffic. This means focusing on the macro level of reliability evaluation rather than on micro-level evaluation, which includes deeper operational characteristics of the whole network.
2) Although there is a rich body of literature that addressed network reliability, there is a lack of reliability research analysis and evaluation in the context of migration from traditional networks to SDN-based networks. Additionally, we employ an edge-disjoint-paths strategy, which is critical to network reliability to avoid message delivery failure due to network resource sharing. It ensures that multiple packet flows from one host node to another host node fully exploit path diversity and do not go through shared links.
3) None of the current proposals provide detailed characteristics of reachability and reliability using a graph operator for a link failure process. A link failure process needs to be defined as a stochastic-based operator that can be employed to study the operational lifetime of a network under any failure distribution and for any network size and topology.

III. Network Modeling

A network is represented as a simple undirected graph $G(V,E)$, where V is the set of vertices/nodes which represent switches, router, terminals/hosts, etc., and E is the set of edges/links that refer to the connectivity between its nodes. Both V and E sets represent the network's elements. Let $n = |V|$ and $m = |E|$ be the number of nodes and links, respectively. Let $Y\ (t) = \{y_1(t),\ y_2(t),\ \ldots, y_m(t)\}$ be a binary vector that represents the current status of links (i.e., link-status vector), where a state $y_i\ (t) =\ 1$ represents that link e_i is available (up/present) at time t; otherwise, link e_i is unavailable (i.e., down/failed). The failure probability of each link is a random variable that takes values according to a given probability distribution. The probability distribution of link failure is usually generated from statistical data. We denote $\Pr(e_i)$ as the probability of link e_i being present, where $\Pr(.)$ is the probability function. Similarly, $H \subseteq V$ denotes the set of all hosts in G, and $K \subseteq \{H \times H\}$ refer to host/terminal pairs that intend to communicate with each other. We denote the size of K by n_k. That is, $K = \{(h_{11}, h_{12}), (h_{21}, h_{22}), \ldots.., (h_{n_k1}, h_{n_k2})\}$, where $n_k = |K|$, and $n_k \leq |H|(|H| - 1)$. Thus, there will be n_k packet streams passing through host pairs. Similar to the link-status vector, we use $X\ (t) = \{x_1(t),\ x_2(t),\ \ldots, x_{n_k}(t)\}$ be a binary vector that represents the status of packet flow delivery at time t (let us call it flow-delivery-status vector), where a state $x_i\ (t) =\ 1$ refers to the packet stream passing host pair (h_{i1}, h_{i2}) at time t is successfully delivered; otherwise, it is failed. A successfully delivered packet stream is transmitted through an operational/up path which is defined as a set of up edges with no edge repetition.

The assumptions made in this article are as follows. 1) A network is represented as a simple graph in which there is no more than one link between any pair of nodes. 2) Sending/receiving data packets occurs between pairs of set K only. 3) Although our approach can be applied on nodes too, we only allow links to fail independently from each other with known probability. 4) For consistent and fair reliability comparison between legacy and SDN-based networks, we assume that the links attached control and data planes do not fail. That is, we focus on the connections of the forwarding nodes. 5) The failed link is not restored and remains down.

IV. Reliability Metric

This section illustrates the reliability metric and its related concepts. Our analysis and formulation are general and independent of the network topology. Studying network reliability is of paramount importance for both the network operator and the end-user. Due to the stringent requirements of services such as real-time services, the network's element failure, such as a link failure, may cause a service to be deemed unreliable. Therefore, the degradation of the network elements has a critical influence on network operation reliability. A reliable network should continue working under augmented link failure. In general, reliability is the probability that a system continues providing its intended functions/services over a period of time and under its operating conditions [47]. In a communication network, a reliability metric measures the robustness of the network after a possible failure of some nodes or links. Thus, providing reliable services by network operator mandates meeting the requirements of these services.

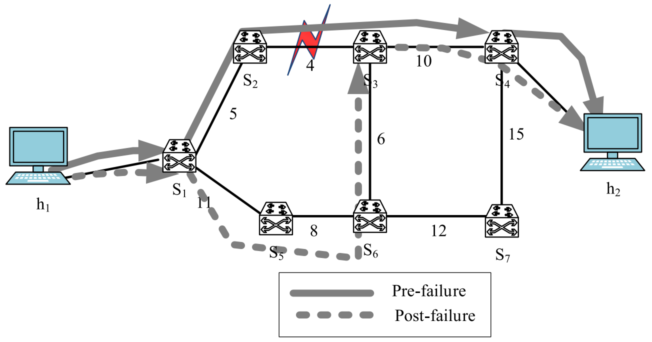

Fig. 1. Illustration of a packet routing within a network before and after a link failure.

In this article, successive link removals (i.e., failure process) is applied to the set of links E. We assume that the vertex set V is fault-free (i.e., failure never occurs to an element in V). We perform reliability analysis and evaluation to test which of the two networks, legacy and SDN-based, will successfully deliver more packet streams per time unit in the presence of a fault. This metric evaluates the ability of each network design to update the routing entries upon link failures quickly. This assumes that the entire network is still connected after a link removal. However, it is possible that the graph G, which represents the network, becomes disconnected after a series of link failures. That is G is fragmented into multiple components or subgraphs, which causes the nodes in these components to no longer be able to communicate with each other. Therefore, regardless of the network design used, legacy, or SDN-based network, the receiver node will never be able to receive any packets from the sender node when sender and receiver nodes are in segregated components.

We differentiate between two cases of unsuccessful delivery of a packet flow upon a link failure: 1) Unreliability due to a network failure to reroute the affected stream in a timely manner (i.e., delay-convergence problem), 2) unreliability due to unreachability as sender and receiver nodes (i.e., host pair) reside in different components. Reliability and reachability/connectivity are tightly coupled in network analysis. While reliability assesses the current status of a network, unreachability examines a possible future network dysfunction. Thus, it is essential not to mix up the two cases. We should first identify the characteristics of both unreachability and reliability before evaluating the reliability metric of the two network design architectures (i.e., SDN-based and legacy). Fig. 1 provides an illustration of a packet routing in a network before and after a link failure. There are two host nodes (h_1, h_2), and seven switches (S_1,.., S_7). Each link has a cost that will be used to establish the shortest path (SP) to route packets. As the figure shows, the communicating host nodes, h_1 and h_2, will still be reachable after the link between S_2 and S_3 failed (i.e., there is an alternative path that connects h_1 and h_2). However, the network reliability depends on the response time of the network, and whether or not this time is within the timeout of the service of interest (i.e., the stringent requirements of a complete convergence of a network upon failure is typically sub-50 ms [48]). That is, the delay of the convergence procedure that runs to recover from a link failure. This procedure includes detecting a failure event, computing an alternative routing path, and restoring the traffic flows.

We recall that the failure process occurs on links only. A similar approach can be followed for nodal failure. Furthermore, link failure is an iterative process and can be defined as an iterated graph operator L, as shown in the following definition.

Definition 1: Let G be a graph that represents a communication network, L is defined as a graph operator $L : G(V, E) \rightarrow G(V, E')$, where E' is obtained by possible removal of one or more links from E. The ith-iterated link removal, L^i, $i \geq 0$, is recursively defined as follows:

$$L^i(G) = \begin{cases} G, & if\ i = 0 \\ \\ L\left(L^{i-1}(G)\right), & otherwise. \end{cases} \quad (1)$$

Definition 1 can be extended to the domain of multi-iterations as follows:

$$L^i(G) = L^j\left(L^{i-j}(G)\right), \text{ where } i \geq j \geq 0, \text{ and } L^0(G) = G. \quad (2)$$

To prove (2), we use the following auxiliary property $L(L^{i-1}(G)) = L^{i-1}(L(G))$ which can be proved by induction; we omit the proof for simplicity. Equation (2) can also be proved by induction as follows. For $j = 0$, $j = 1$, and $j = i$, equation (2) holds directly. Let us prove it for $i > j > 1$. We assume that (2) is true for $j = i - 2$. We prove it for $j = i - 1$.

$$\begin{aligned} L^i(G) &= L^{i-1}\left(L^{i-(i-1)}(G)\right) = L\left(L^{i-2}(L(G))\right) \\ &= L^{i-2}(L(L(G))) = L^{i-2}\left(L^2(G)\right). \end{aligned}$$

Equation (2) holds a transitivity property: If a flow-delivery-status vector $X(i)$ has been produced through some iterations of L on network G, $G_i = L^i(G)$ and $X(r)$ has been obtained through some iterations on network G_i, then $X(r)$ can be obtained through some iterations of L on network G.

This property helps to predict and maintain reliability over the network's lifetime. The failure probability distribution of links in a graph G determines which links will be removed in each iteration and their removal order. Therefore, L could remove zero or more links in any iteration.

A. Unreachability Characterization

In this subsection, network unreachability analysis is conducted by successive link removals while keeping all other parameters such as the number of nodes, link failure rates, and link bandwidths unchanged. This type of analysis shows the network unreachability characteristics over time. Thus, it helps find the bottleneck of the network operational lifetime of sending and receiving packets. In this analysis, we create a network, apply a failure operator L, and then observe the graph dynamicity over time and its effect on nodal reachability. We provide the following auxiliary graph notations to illustrate the unreachability.

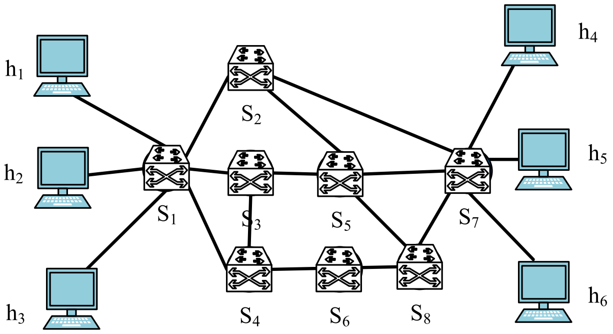

Fig. 2. Legacy network of 14 nodes and 18 links.

1) A graph G is connected when there is a path (i.e., a sequence of links) that connects any two nodes in G.
2) A degree of node v in a graph G, $\deg(v)$, is the number of links attached to v.
3) The cut-edge set, $e\_cut(G)$, of a connected graph G is a set of links whose removal disconnects G. This set is not unique. The minimum cut-edge set, $min\_e\_cut(G)$, is the cut-edge set with the smallest number of links. For any connected graph G it holds that

$$|min\_e\_cut(G)| \leq min\_deg(G) \quad (3)$$

where $|.|$ denotes the size of a set, $min\_deg(G)$ is the minimum node degree in G. A graph G is called k-connected if $k = |min\_e\_cut(G)|$. When a network becomes disconnected, the disruption in packet flow delivery can be assessed by measuring the number of fragmented components and their sizes (i.e., the number of nodes in each component). The more components, the more degradation in delivering packet flows and, hence, the more unreachability among the network's elements. The principle of minimum cut-edge sets plays an important role in nodal unreachability.

Fig. 2 shows a network of 14 nodes (6 hosts $(h_1, .., h_6)$, 8 switches $(s_1, .., s_8)$) and 18 links. Suppose the link failure probability follows a uniform distribution. We assume that operator L removes one link randomly in every iteration. To avoid possible early isolation of hosts, we also assume that the failure process applies only to the core network, which means link removal will not affect the links between host nodes and their direct switches. Thus, L will run for 12 iterations to uniformly fail the 12 links that are attached to the nodes with labels of s_i, where $1 \leq i \leq 8$.

We use Fig. 2 as a base for the results in Fig. 3. In Fig. 3, we adopt an incremental and irreversible failure process in which link failure with a random uniform distribution is considered. We evaluate two metrics: The number of components and the size of the largest component. The size of the largest component is also known as the relative size of the largest connected component (rLCC) [49], [50]. A cross-comparison between the two curves in Fig. 3 provides a way to analyze features of unreachability under the execution of L. Fig. 3 shows that the graph G remains connected (i.e., one component) until time equals 5. However,

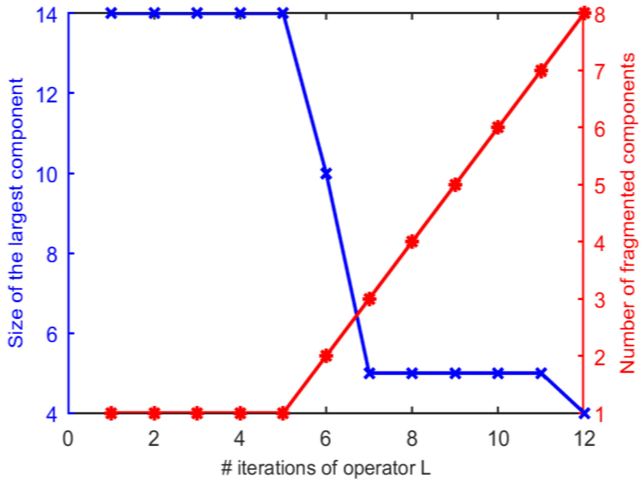

Fig. 3. Number of components and the size of the largest component under the iterations of operator L where a link failure follows a uniform distribution.

the sharp decline in the size of the largest component can be interpreted as more components are created. For example, at time $i = 6$ and $i = 7$, the size of the largest component declined to 10 and 5, respectively. On the other hand, the number of components was only 2 and 3, respectively. This shows that the link that was removed at $i = 7$ was a cut edge of the graph $L^6(G)$. This analysis leads to the following proposition.

Proposition 1: Let G be a network. If a set K of host pairs is reachable at $L^i(G)$, then it is also reachable at $L^{i-j}(G)$, where $i \geq j > 0$.

We omit the proof of proposition 1, which is straightforward. Proposition 1 will be used later in this section to analyze the reliability and reachability.

B. Reliability Characterization

The results of the previous subsection suggest two cases to evaluate network reliability: a) The graph has only one component, and b) the graph is fragmented and has multiple components. In the former case, the reliability assessment is straightforward as each host pair obviously belongs to the same component. However, in the latter case, a test is required to ensure that each host pair resides in the same component to be able to communicate. That is, there should be a path between a source and destination nodes.

The evaluation of a reliability metric of a communication network requires sending packet streams between host pairs and then an assessment of their delivery. We recall that the flow-delivery-status vector at iteration i is $X\ (i) = (x_1(i),\ x_2(i), \ldots,\ x_{n_k}(i))$, where $n_k = |K|$ and K is the set of host pairs that intend to communicate with each other. $x_j(i)$ is a binary random variable that represents the state–of-packet-flow delivery between the jth host pair (h_{j1}, h_{j2}) at iteration i, $1 \leq j \leq n_k$, $x_j\ (i) = 1$ for successful delivery and 0 for failure.

A network G becomes operationally interrupted after c iterations. That is, the set of links to be removed at iteration $(c+1)$ is a cut-edge set of $L^c(G)$.

TABLE I
3-State Vector of Packet Streams as the Number of Iterations i Increases

Iteration# (i)	# Links removed so far	flow-delivery-status vector, $X(i)$, (1 for successful delivery, 0 otherwise)		
		$h_1 \rightarrow h_6$	$h_2 \rightarrow h_5$	$h_3 \rightarrow h_4$
1	1	1	1	1
2	2	0	1	1
3	3	1	1	1
4	4	1	1	1
5	5	1	1	1
6	6	0	0	0
7	7	0	0	0
8	8	0	0	0
9	9	0	0	0
10	10	0	0	0
11	11	0	0	0
12	12	0	0	0

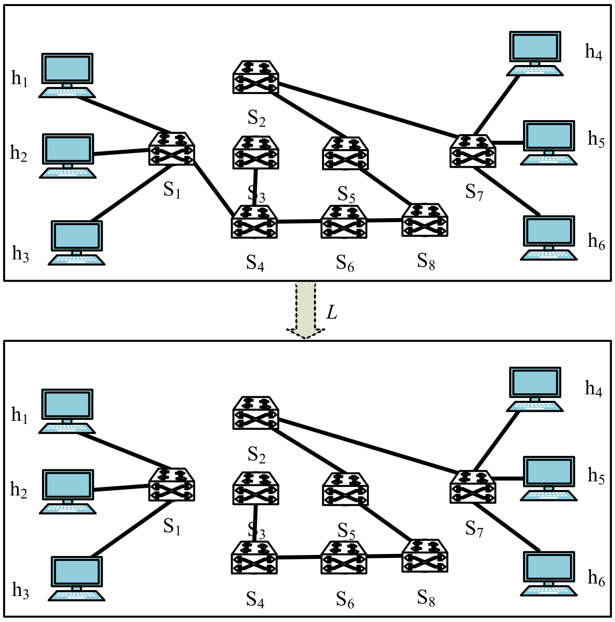

Fig. 4. Two iterations of operator L on graph G: $L^5(G)$ and $L^6(G)$ from top to bottom, respectively. The removal of the link between s_1 and s_4 in $L^5(G)$ leaves $L^6(G)$ disconnected with two components.

Example 1: Table I shows an instance of considering $L^i(G)$ for $1 \leq i \leq 12$. Assume three packet streams with host pairs of set $K = \{(h_1,\ h_6),\ (h_2,\ h_5)\ ,\ (h_3,\ h_4)\}$. We aim to calculate the successful delivery of each packet stream under failures over time. The 3-state vector of all packet streams at iteration i is shown as $X(i) = (x_1(i),\ x_2(i),\ x_3(i))$. For instance, at iteration $i = 5$, $X\ (5) = (1,\ 1,\ 1)$.

In Table I, $L^i(G)$ for $1 \leq i \leq 5$ did not impact the reachability of host pairs. However, the whole network can no longer resist the level of fragmentation it encounters and collapses after time $i = 5$. It also shows one reliability issue recorded at $i = 2$, where the delivery of packet stream between the host pair (h_1, h_6) was failed.

The structures of $L^5(G)$ and $L^6(G)$ are depicted in Fig. 4. A cross-comparison between Table I and Fig. 4 shows that the operational lifetime of a network G ended suddenly after

$i = 5$ (i.e., $X\ (i) = (0,\ 0,\ 0)$ for $i > 5$). This suggests that the link removed from $L^5(G)$ is a cut edge. Tracking the structural change of graph $L^5(G)$ shows that the removal of the link $s_1\ s_4$ leaves $L^6(G)$ dysfunctional.

Reliability Formulation

We assume that each iteration of operator L is performed in a one-time unit. Therefore, we use a one-time unit and one iteration to indicate the same meaning. Consequently, reliability can be represented as a function of time, namely, $R(t)$. We recall that the maximum number of successfully delivered packet streams among all host pairs in K is equal to n_k.

Let $n_{k,s}(t)$ and $n_{k,f}(t)$ refer to how many packet streams were successfully delivered and how many were not, respectively. $R\ (t) = \frac{n_{k,s}(t)}{n_k}$, and $F\ (t) = \frac{n_{k,f}(t)}{n_k}$ denote the probability of survivals and failures of packet streams. Hence, $R(t) + F\ (t)$=1. For instance, if a state vector of packet streams at time t is $X\ (t) = (x_1(t),\ x_2(t),\ x_3(t))$, $R\ (t) = \frac{x_1(t)+\ x_2(t)+\ x_3(t)}{n_k}$, and $F\ (t) = \ 1 - \frac{x_1(t)+\ x_2(t)+\ x_3(t)}{n_k}$. As a numerical example, Table I shows that $n_k = \ 3$ (i.e., three host pairs in set K). At $t = 2$, the state vector is $X\ (2) = (0,\ 1, 1)$ which indicates the successful delivery of two packet streams, i.e., $n_{k,s}\ (2) = \ 2$, and hence, $n_{k,f}\ (2) = \ 1$. Therefore, network reliability $R\ (2) = \frac{2}{3}$, and $F\ (2) = \frac{1}{3}$.

The reliability of a communication network is usually related to the connectivity of the network. We can formulate reliability at time t as follows:

$$R\ (t) = \frac{\gamma\,(t)}{n_k} \tag{4}$$

where $\gamma(t)$ denotes the number of connected/reachable host pairs at time t. In the reliability literature, the average two-terminal reliability (ATTR) refers to the fraction of the number of occurrences the network remains connected over all simulation runs [51], [52]. Two host pairs are connected/reachable when there is a path that attaches them together. On the other hand, Table I shows that at $t = 3$, $x_1\ (3) = \ 1$ which means, by proposition 1, that the connectivity condition was held for the host pair corresponding to $x_1(2)$ (i.e., h_1 and h_6 should be connected at $t = 2$), yet the packet stream was not delivered as $x_1\ (2) = \ 0$. This is because there are other factors involved so that traffic flow was not delivered, although a path between the two hosts was present. Potential micro-level factors could be related to active queue management (AQM), scheduling, buffering, rerouting, bandwidth management, etc. Let us denote these factors as resource-related traffic factors (RTF). Thus, the competition on shared network resources such as the capacity of links and switches between different packet flows contributed to the degradation of network reliability.

Example 2: Using Fig. 2, assume two pairs of terminals $(h_{i1}, h_{i2}) = (h_1,\ h_4)$, and $(h_{j1}, h_{j2}) = (h_2,\ h_5)$. $(h_1,\ h_4)$ is connected through a path $p_i = \{h_1,\ s_1,\ s_3,\ s_5, s_7,\ h_4\}$, while $(h_2,\ h_5)$ is connected through a path $p_j = \{h_2,\ s_1,\ s_3,\ s_4, s_6,\ s_8,\ s_7,\ h_5\}$. p_i, and p_j share at least one network element, namely, the two nodes s_1 and s_3 and the link between them, e. If any factor from RTF occurs at time t, then at least one of the packet streams through these paths may not be delivered, i.e., $x_i\ (t) = \ 0$ or $x_j\ (t) = \ 0$, and, hence, $R(t) < 1$.

Clearly, a communication network has no reliability when every host pair in set K is disconnected, i.e., $\gamma\ (t) = \ 0$. However, what is more interesting is that reachability is not enough for reliability. Indeed, resource availability for packet messages should be guaranteed in order to guarantee reliability. Link congestion is one of the most critical reasons behind packet loss [53]. As a concrete input for the above example, if the shared link e at time t did not have enough bandwidth to allow both packet messages passing through p_i, and p_j, then degradation of reliability would be an issue in this case. Therefore, adopting edge-disjoint routing among host pairs would allow traffic to flow smoothly. We address the impact of disjoint paths on reliability in Section IV-D.

Calculating the average reliability at time t over multiple instances of $L^i(G)$ can be achieved by applying the whole set of link removal process multiple times, e.g., θ times, as follows:

$$R_{Avg}\ (t) = \frac{1}{\theta}\ \sum_{k\,=\,1}^{\theta} R_k\ (t) \tag{5}$$

where $R_k(t)$ is $R(t)$ at instance k.

Proposition 2: Let $G(V, E)$ be a network. Considering (4) and with the absence of RTF's impact, the network reliability of G under application of operator L is a nonincreasing function in time. That is $R(t) \geq R(t + \Delta)$, where $\Delta\ \geq 0$.

Proof: Let $F(t)$ and $f(t)$ be the failure cumulative distribution function (CDF), and the failure probability density function (PDF), respectively, of links applied by operator L on G. Thus, $R\ (t) = \ 1 - F\ (t) = \ 1 - \int_0^t f(t)dt$. At time $t + \Delta$, $R\ (t + \Delta) = \ 1 - \int_0^{t+\Delta} f(t)dt$.

Therefore, the change in network reliability in a period of time $[t, t + \Delta]$ is

$$\begin{aligned} \mathrm{R}\,(t + \Delta) - R\ (t) \ &= \ -\int_0^{t+\Delta} f\,(t)\,dt + \int_0^t f\,(t)\,dt \\ &= \ -\int_t^{t+\Delta} f\,(t)\,dt. \end{aligned}$$

■

Proposition 2 shows that reliability under a failure process is a monotonic decreasing function in time. However, this is guaranteed under the absence of RTF's impact. In fact, this condition can be relaxed a little bit to read "under a systematic impact of RTF" instead of a complete "absence of RTF's impact." In other words, if the impact of RTF is consistent at any time t, then reliability under operator L will behave as a none-increasing function over time. Based on this proposition, we have the following corollary.

Corollary 1: Let $G(V, E)$ be a network. Considering the absence of RTF, the average network reliability under the application of operator L is a nonincreasing function in time t. That is $R_{\mathrm{Avg}}(t) \geq R_{\mathrm{Avg}}(t + \Delta)$, where $\Delta\ \geq 0$.

Proof: Follows from (5) and Proposition 2. ■

In this article, we also use the following indicators to quantitatively analyze the operational reliability: Expected packets

TABLE II
CALCULATION OF THE INDICATORS BASED ON TABLE I

Failure's time t	$epnd(t)$	$ppopl(t)$	$npidr(t)$
1	0	0	0
2	1	1	0.33
3	0	0	0
4	0	0	0
5	0	0	0
6	3	1	1
7	3	1	1
8	3	1	1
9	3	1	1
10	3	1	1
11	3	1	1
12	3	1	1

not delivered (EPND), the presence probability of packet loss (PPoPL), and the network packets interruption duration ratio (NPIDR). Let T be the time interval of each failure iteration, and T_{sum} be the whole iterations time. The indicators *epnd*(t), *ppopl*(t), and *npidr*(t) at time t can be calculated as follows:

$$epnd\ (t) = \ T \times [P_s\,(t) - P_d\,(t)] \tag{6}$$

$$ppopl\ (t) = \begin{cases} 1\ if\ epnd\,(t) > 0 \\ 0\ otherwise \end{cases} \tag{7}$$

$$npidr\ (t) = \frac{epnd\,(t)}{P_s\,(t)} \tag{8}$$

where $P_s(t)$ is the number of packets sent and of which a number of $P_d(t)$ is successfully delivered. The indicator *epnd*(*t*) represents packets undelivered in each simulated failure at time t. *ppopl*(*t*) shows the presence of the undelivered packets in each simulated failure iteration *t*, one for the existence and zero otherwise. And the indicator *npidr*(*t*) describes the duration of the undelivered packets per failure iteration. Based on the 3-state-vector in Table I, we calculate the three indicators as shown in Table II.

After the iterations are stopped, the network reliability indicators can be calculated by summing up their values over all iterations (i.e., T_{sum}). Thus, $EPND\ = \frac{22}{12}\ = \frac{11}{6}$ packets per iteration, $PPoPL\ = \frac{8}{12}\ = \frac{2}{3}, NPIDR\ = \frac{7.33}{12}$ per iteration time.

C. Impact of Link-Failure Probability Distribution on Network Reliability

Equation (4) calculates reliability at time t by counting the number of successfully delivered messages over the total number of messages. However, a general analysis of network reliability requires prior knowledge of the link failure probability distribution, which can be generated from historical statistical data. Therefore, failure probability distribution used by operator L to remove links from network G has a direct impact on a link-state vector $Y(t)$ and, consequently, it affects the traffic-delivery-state vector $X(t)$. As there are m links in a network G, $Y(t)$ has 2^m possible combinations/settings; each of which will directly affect $X(t)$. Thus, the reliability of G will now be related to each setting of $Y(t)$. Equation (4) can be employed as follows:

$$R\ = \sum_{t=1}^{2^m} R\,(t)\,\Pr\,(Y\,(t))\ = \sum_{t=1}^{2^m} \frac{\gamma\,(t)}{n_k}\,\Pr\,(Y\,(t)) \tag{9}$$

where

$$\Pr\,(Y\,(t)) = \prod_{\substack{1 \le k \le m \\ y_k\,(t) = 1}} \Pr\,(e_k) \prod_{\substack{1 \le k \le m \\ y_k\ (t) = \ 0}} (1 - \Pr\,(e_k))\,. \tag{10}$$

We recall that a state $y_k\ (t) = \ 1$ represents that a link e_k is up/present at time t; otherwise, link e_k is down/failed. The computation cost to evaluate all settings grows exponentially in the number of links. In fact, the evaluation of reliability is proved to be a #P-complete problem [9], [10], and therefore reliability metric cannot be computed for large-scale networks through link-state vector combinations.

D. Impact of Disjoint Paths on Reliability

The discussion and analysis in Section IV-B showed that the existence of reachability does not guarantee network reliability. It suggests the criticality of disjoint resources besides maintaining reachability to mitigate any possible degradation in reliability due to the presence of RTF. That is, disjoint routing paths do not compete for network bandwidth, reduce congestion, and enhance network reliability [54]. Furthermore, in case of a link failure, path disjointness presents a good solution to avoid rerouting traffic flow through the path associated with the failed link. In this subsection, we investigate packet message delivery via disjoint paths.

Paths could be either disjoint via vertices or edges. However, disjoint paths via disjoint vertices (vertex-disjoint paths, VDP) is a superset of disjoint paths via disjoint edges (edge-disjoint paths, EDP). That is EDP $\subseteq$ VDP. In this article, we focus on EDP. Menger's theorem [55] states that in a graph G the size of a minimum cut-edge set, i.e., $min\_e\_cut\ (G)$, is equal to the maximum number of disjoint paths that can be found between any host pair. Using (3) the size of EDP for each host pair is given by the following:

$$|\text{EDP}| \le min\_deg\,(G) \le n - 1. \tag{11}$$

Let (h_{i1}, h_{i2}), where $i \in \{12, .., n_k\}$ be a host pair of K that are connected through disjoint paths $D_i = \ \{p_1,\ p_2, \ldots,\ p_r\}$, where $r\ = |\text{EDP}|$ refers to the size of the set of edge-disjoint paths. With a proper link-labeling, D_i can be calculated by running a breadth-first search r times. This yields a complexity of $O(r(m + n))$. We order these paths, such that $|p_1| \le |p_2| \le$ $, \ldots, \le |p_\tau|$, where $|p_j|$ refers to the number of edges/hops that form p_j. A path p_j is operational, p_{j_opl}, when all edges constituting p_j are up/present. The probability of p_{j_opl} is formulated as follows:

$$\Pr\,(p_{j\_\text{opl}}) = \prod_{\substack{1 \le k \le |p_j| \\ e_k \in p_j}} \Pr\,(e_k). \tag{12}$$

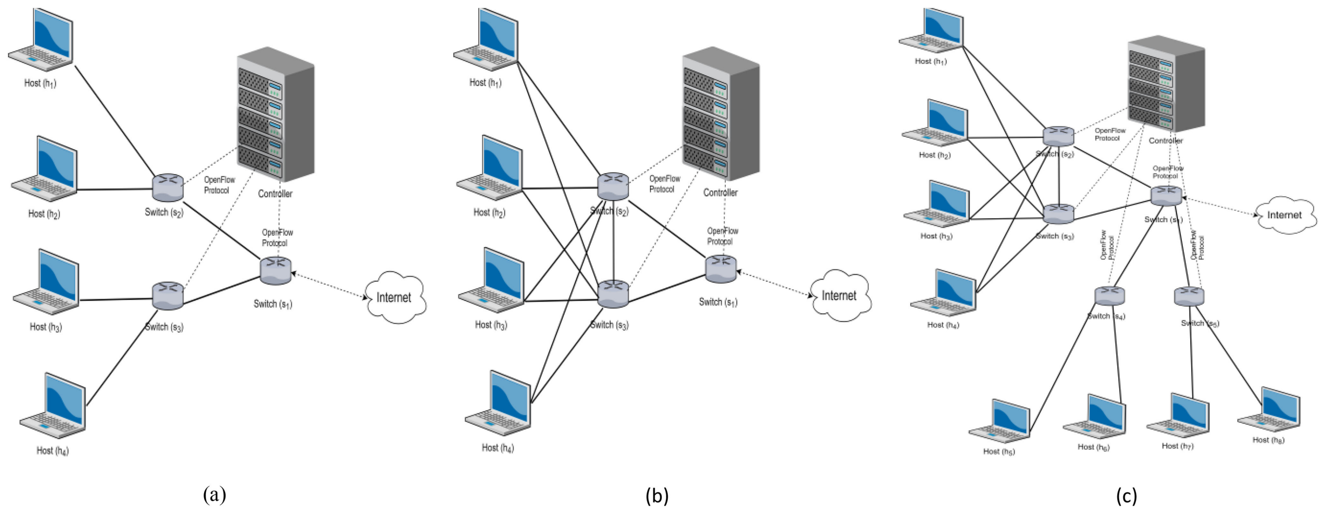

Fig. 5. SDN-based network topologies. (a) TREE(Original). (b) MESH(Original). (c) HYBRID(Original).

We assume that the routing of packet traffic follows a shortest-path method. This means the probability that p_j is the shortest path to be used, p_{j_usd}, by (h_{i1}, h_{i2}) is given by

$$\Pr(p_{j\_\text{usd}}) = \prod_{z=1}^{j-1} (\Pr(p_{z\_fa})) \,.\, \Pr(p_{j\_\text{opl}}) \tag{13}$$

where p_{z_fa} denotes the path p_z is failed, which has the following probability:

$$\Pr(p_{z\_fa}) = \prod_{z=1}^{j-1} \left(1 - \prod_{\substack{1 \le k \le |p_z| \\ e_k \in p_z}} \Pr(e_k) \right). \tag{14}$$

Therefore, network reliability under edge-disjoint can be formulated as follows:

$$\begin{aligned} R &= \sum_{\substack{i=1 \\ (h_{i1}, h_{i2}) \in K}}^{n_k} R(t) \sum_{\substack{j=1 \\ p_j \in D_i}}^{r} \Pr(p_{j\_\text{usd}}) \\ &= \frac{1}{n_k} \sum_{\substack{i=1 \\ (h_{i1}, h_{i2}) \in K}}^{n_k} \sum_{\substack{j=1 \\ p_j \in D_i}}^{r} \Pr(p_{j\_\text{usd}}) \,. \end{aligned} \tag{15}$$

Based on (15), a simple and efficient algorithm can be designed to calculate reliability under edge-disjoint paths. We omit the algorithm since it is straightforward. However, we show its computational efficiency in the next proposition.

Proposition 3: K-pair reliability of a network G can be calculated by (15) in polynomial time of $O(n(m+n)n_k)$.

Proof: The time complexity of the omitted algorithm can be calculated as follows. The outer loop (i.e., the outer summation) iterates a number of times equal to the size of a host-pair set of K (i.e., n_k). In each iteration, finding D_i requires a time of $O(r(m+n))$. The number of iterations of the inner loop is r which upper-bounded by $n-1$, according to (11). The calculation of $\Pr(p_{j\_usd})$ requires $O(n)$. Therefore, the total time complexity is $O((n_k(m+n) + n_k n)\ (n-1)) = O(n(m+n)n_k)$. ■

V. Network Topology and Performance Evaluation

In order to cover most types of network topologies, we adopt a tree, mesh, and hybrid network topologies, as shown in Fig. 5. We omit the figures of legacy network topologies because they are like their SDN-related network counterparts, except they do not have an SDN controller.

A. *Emulation and Simulation Environment*

SDN-enabled network topology of our experiment is emulated using Mininet 2.2.2 [56] with Open vSwitch (OVS) 2.5.4 supported by OpenFlow version 1.3 [57]. Mininet can be used as both an emulator and a simulator [24], [58]. It is widely used by many researchers for performance evaluation of SDN-based networks in different domains such as acoustic sensor networks, communication networks, and security [59]–[63]. A management problem of the physical resources such as CPU power and memory can lead to bottlenecks in the Mininet environment, especially in large-scale emulated networks [64]. In our experiments, we use small-size topologies. Furthermore, we employ the flexibility of Mininet to allocate fractions of the resources of the host machine such as CPU power, memory allocation, etc. For example, we manage CPU power through "mininet.node.CPULimitedHost Class" to assign a fraction of CPU power for each emulated host node (i.e., VM) [65]. Similarly, the characteristic parameters of each virtual link such as bandwidth, packet loss, delay, and jitter can be assigned a specific value. Mininet also creates unique network namespaces to isolate the communications between the VMs and the host system. Therefore, our experiments are conducted in a dedicated, uninterrupted, and stable platform.

In our Mininet setting, we follow the same approach as in [24]. Based on the three types of network topologies in Fig. 5, 12 network topologies were generated and considered in our study as will be shown in Section V-B. These topologies have a various number of network elements with maximum and (minimum) values of 32(4) hosts, 17(3) switches, and 68(6) links, respectively. In the emulation, each host node uses the pingAll() function to send the ECHO_REQUEST message to test the connectivity with all other host nodes. ECHO_REPLY packets are captured at the sender's host using the tcpdump tool. Links are configured with a 1-ms delay. The size of ICMP Ping message is 64 bytes [66]. The time between every consecutive ping is determined by response time to receive the ECHO_REPLY, which, in turn, depends on the length of the path used to transmit the packet. In the worst-case scenario, the round-trip time (RTT) is 136 ms $(= \ 68(links) \times 1(ms) \times 2(round\ trip))$, while it is 12 ms in the best case scenario [i.e., the case of tree topology in Fig. 5(a)]. Accordingly, the bandwidth of traffic flow between host pairs is no more than 41.6 kbps (i.e., $(64 \times 83.3 \times 8)/1024$). We set up a capacity of each link at 1000 Mbps as in [24].

Using Mininet, we run a Python program agent to simulate traffic flows in each network. Furthermore, Floodlight [67] V1.2 is used as the SDN controller for all experiments. Floodlight is based on the Java programming language. It adopts a forwarding policy that uses the shortest-path tree to forward data flows for every switch [68]. Since the size of our network topologies is relatively small, one controller would be enough in the simulation of all experiments.

On the other hand, we follow the same settings to emulate and simulate legacy network topologies. However, to mimic legacy switches, we set OVS into standalone and spanning tree protocol (STP)-enabled mode, and, of course, there is no controller in this case.

These applications (e.g., Mininet, Floodlight, and bash shell) run on ThinkCentre Desktop with Intel Core$^{\text{TM}}$ i5-6500T CPU @ 2.5GHz, 8GB of RAM and 64-bit Ubuntu 16.04LTS operating system.

The all-terminal reliability-evaluation problem is computationally intractable for arbitrary networks and shown to be #P-complete [9][10]. In all-terminal reliability-evaluation research problem, using limited-size network topology (such as a network with 68 links or less) has been widely adopted in several research papers [69]–[74] for validating their solutions.

B. Experiments and Performance Evaluation

We will present three sets of reliability evaluation, according to the various topologies used: Tree-based, Mesh-based, and Hybrid-based. Furthermore, each of which will have three extensions to provide a wide range of topologies within the same type but different in network size, and the number of hierarchal levels. Hence, we consider the following parameters.

1) The number of levels (i.e., depth) of each network type is 1, 2, 3, or 4, where depth 1 refers to the original graphs depicted in Fig. 5. Each additional level will increase the number of switches and hosts (i.e., network size) by 100% of the original network excluding the root switch, s_1. For example, a two-level topology of a tree type will include one more level compared to the original graph. This will double the number of links and hosts. However, the number of switches (#S) will increase by two as shown in Table III. Therefore, we call the Two-level tree topology simply by TREE(Two-level). The same analogy will apply to other topologies.

TABLE III
COMPARISON OF DIFFERENT PARAMETERS BETWEEN VARIOUS TYPES OF TOPOLOGIES AND THEIR VARIATIONS

Structure Type		Original	Two-level	Three-level	Four-level
TREE	#L	6	12	18	24
	#S	3	5	7	9
	#H	4	8	12	16
MESH	#L	11	22	33	44
	#S	3	5	7	9
	#H	4	8	12	16
HYBRID	#L	17	34	51	68
	#S	5	9	13	17
	#H	8	16	24	32

2) The pattern of connecting switches to switches and switches to hosts will be the same. For example, the tree-type network will connect each switch with only two hosts.

Table III shows the comparison between the three topologies against the number of links (#L), the number of switches (#S), and the number of hosts (#H).

First, we present the reliability-related settings before addressing the performance evaluation of the various topologies. In our experiments, we adopt the graphs G in Fig. 5(a)–(c) and their extensions, as described in Table III. Legacy network topologies are similar to their SDN-based counterparts but without a controller. This analysis aims to measure the average $all-pair-reliability$ of each network design under the iterated failure process using operator L. To achieve this, for each iteration i, the following steps will take place: 1) Apply link removal operator $L(G)$, 2) generate packet flows by using the pingAll() function. Hence, test connectivity between each host pairs $(h_i,\ h_j)$, where $1 \leq i, j < n,$ and $i \neq j$, and then 3) calculate the traffic-delivery-state vector $X(i) = (x_1(i),\ x_2(i), .., \ x_{n_k}(i))$, where $n_k = |H|(|H| - 1)$. The steps 1), 2), and 3) will be repeated until one link remains unremoved or reliability reaches zero. We calculated $R_{\text{Avg}}(t)$ for different trials (1, 3, 6, 10, 15, 20, 25, 40). We found that the result is convergent. It always converges and stabilizes after 15 trials. Our measurements of the variance of $R_{\text{Avg}}(t)$ suggested that $\theta = 20$ should provide stable and convergent results. Therefore, the whole experiment is repeated 20 times for each network topology.

We consider the exponential distribution as a lifetime model for our networks' links. That is $R\ (t) = e^{-\lambda t}$, where λ is a scale parameter (also known as a hazard rate). The exponential distribution is widely used by the reliability research community as a time-to-failure model for various systems.

In exponential distribution, the failure of an element of the network may not be due to aging, but rather to random events related to network operation such as communication or routing

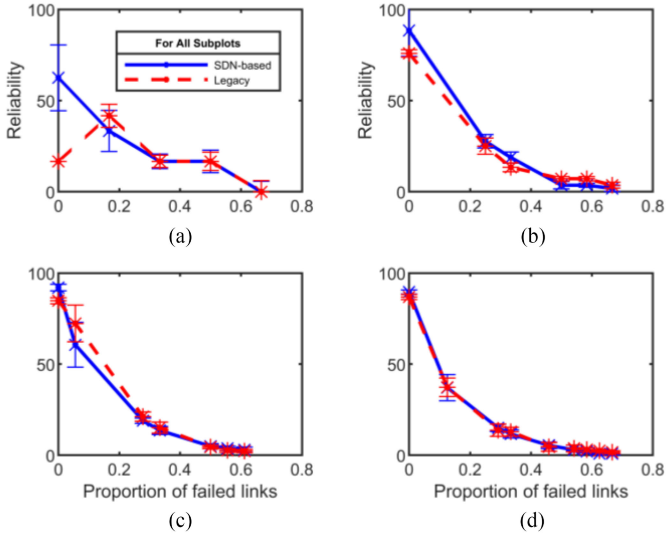

Fig. 6. Reliability over time of SDN-based versus legacy tree-topologies as the failure process follows an exponential distribution with λ = 1/4.

faults. The link removal operator $L^i(G)$ follows a link failure probability which is defined as [75]:

$$\Pr(0 \leq \mathrm{T} \leq \mathrm{t}) = 1 - \int_{\mathrm{t}}^{\infty} \lambda \mathrm{e}^{-\lambda \mathrm{t}} \mathrm{dt} \;=\; 1 - \mathrm{e}^{-\lambda \mathrm{t}},$$

where T is a random variable that refers to a lifetime of a network link (i.e., representing the time until a link fails).

Next, we present our experiments to evaluate the performance evaluation of the two systems against operational reliability and under three sets of topologies: Tree, mesh, and hybrid, along with their extensions. Each set deals with the performance evaluation of one of the three topologies against the reliability metric. The results of each set will be presented as a grid of four subfigures associated with the four level-based topologies: Original, Two-level, Three-level, and Four-level as shown in Table III.

Set 1: Reliability under Tree-based Topologies

Fig. 6 shows the performance evaluation that is conducted for the tree network type which Original/One-level topology depicted in Fig. 5(a). The subfigures of Fig. 6 show the performance of reliability as a link failure process proceeds (proportion of failed links). We recall that each point in every graph is averaged from 20 replication scenarios. The performance of the two designs is comparable with SDN-based network outperforming the original and the two-level topologies in some cases. It should be noted that at around 70% of the accumulated failed links, the reliability values of both architecture designs approach zero for all the tree level-based topologies.

Set 2: Reliability under Mesh-based Topologies

In this set, performance is evaluated for the mesh-type network. Like the tree-type network, the performance of the two architecture designs is similar, as shown in Fig. 7. However, there is a slight outperformance of SDN over legacy in MESH(Three-level).

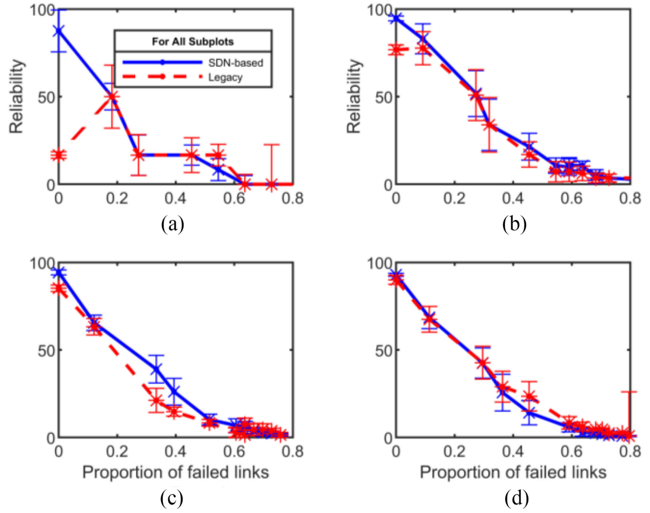

Fig. 7. Reliability over time of SDN-based versus legacy mesh-topologies as the failure process follows an exponential distribution with λ = 1/4.

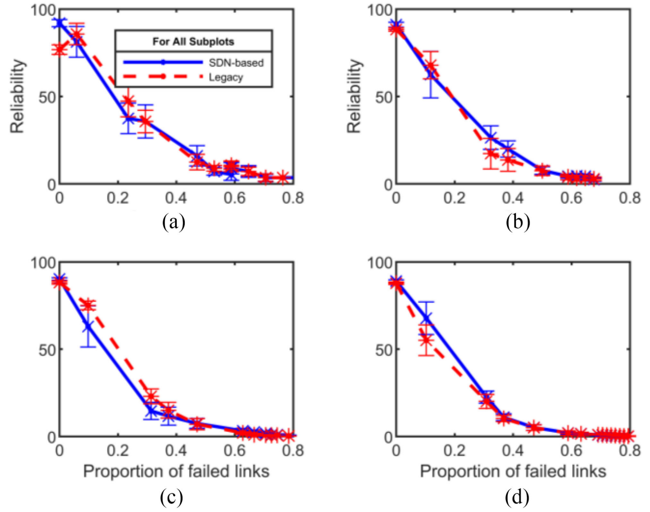

Fig. 8. Reliability over time of SDN-based versus legacy hybrid-topologies as the failure process follows an exponential distribution with λ = 1/4.

We observe that legacy networks lack fast routing reconfiguration at the start of each experiment (i.e., $x = 0$). This is because the legacy network does not know the global topological information which can be used as leverage for making fast routing decisions. However, it is more apparent in topologies with a small number of hosts. For example, TREE(Original) in Fig. 6 and MESH(Original) in Fig. 7 have both four hosts, which is a relatively small number compared to the other topologies. Consequently, there is a gap in performance at the beginning of each experiment.

Set 3: Reliability under Hybrid-based Topologies

Set 3 is evaluated for hybrid-type networks. Fig. 8 shows the performance of both systems, which is not different from what we observed in Figs. 6 and 7. The performance of legacy networks is close to SDN-related networks.

Figs. 6–8 show that the majority of the visible differences in reliability values occurred at $x = 0$ (i.e., at zero proportion

TABLE IV
EVALUATE THE STATISTICAL SIGNIFICANCE OF THE DIFFERENCES IN RELIABILITY VALUES AS PRESENTED IN FIGS. 6–8 USING T-TEST. RESULT IN EACH CELL IS REPRESENTED AS FOLLOWS: H/P ON THE TOP AND THE CONFIDENCE INTERVAL BELOW IT. h IS EITHER 1 OR 0 (REJECT OR ACCEPT) REJECT THE NULL HYPOTHESIS, WHILE P INDICATES THE P-VALUE

Iteration # Topology	1	2	3	4	5	6	7	8	9
TREE(Original)	0/0.064 (-2.44, 0.77)	0/0.509 (-7.14, 3.81)	0/0.509 (-7.14, 3.81)	0/0.343 (-10.87, 4.20)	0/0.343 -10.87, 4.20)	0/0.678 (-10.46, 7.13)	0/0.678 (-10.46, 7.13)	0/0.678 (-10.46, 7.13)	0/0.678 (-10.46, 7.13)
TREE(Two-level)	0/0.237 (-6.17, 2.88)	0/0.716 (-5.39, 7.54)	0/0.223 (-1.56, 5.85)	0/0.298 (-3.81, 1.31)	**1/0.003 (-5.10, -1.33)**	**1/0.011 (-3.78, -0.69)**	1/0.011 (-3.78, -0.69)	1/0.011 (-3.78, -0.69)	0/0.140 (-5.40, 0.94)
TREE(Three-level)	0/0.334 (-2.12, 9.12)	0/0.310 (-4.94, 1.75)	0/0.360 (-5.83, 2.35)	0/0.714 (-1.51, 2.11)	0/0.274 (-0.93, 2.90)	0/0.274 (-0.93, 2.90)	0/0.145 (-0.54, 3.11)	0/0.145 (-0.54, 3.11)	0/0.145 (-0.54, 3.11)
TREE(Four-level)	0/0.804 (-9.80, 2.30)	0/0.622 (-2.86, 4.53)	0/0.607 (-3.28, 2.03)	0/0.635 (-2.80, 1.80)	**1/0.042 (-3.34, -0.07)**	**1/0.042 (-3.34, -0.07)**	0/0.051 (-2.32, 0.01)	0/0.051 (-2.32, 0.01)	0/0.222 (-1.59, 0.42)
MESH(Original)	0/0.531 (-6.11, 4.44)	0/0.745 (-9.37, 1.37)	0/0.504 (-1.72, 9.38)	0/0.678 (-1.46, 7.13)	0/0.678 (-1.46, 7.13)	0/0.678 (-1.46, 7.13)	0/0.678 (-1.46, 7.13)	0/0.678 (-1.46, 7.13)	0/0.678 (-1.46, 7.13)
MESH(Two-level)	0/0.581 (-8.95, 1.02)	0/0.890 (-8.68, 2.18)	0/0.954 (-2.03, 9.96)	0/0.724 (-9.30, 1.87)	0/0.726 (-7.52, 1.38)	0/0.726 (-7.52, 1.38)	0/0.930 (-8.63, 9.35)	0/0.510 (-6.05, 1.21)	0/0.510 (-6.05, 1.21)
MESH(Three-level)	0/0.06 (-0.13, 5.28)	0/0.179 (-3.68, 7.01)	**1/0.037 (0.76, 9.24)**	0/0.122 (-1.18, 8.45)	0/0.207 (-2.11, 8.47)	0/0.232 (-1.98, 7.03)	0/0.697 (-3.57, 5.09)	0/0.945 (-3.56, 3.78)	0/0.945 (-3.56, 3.78)
MESH(Four-level)	0/0.730 (-9.13, 1.54)	0/0.431 (-1.54, 7.71)	0/0.558 (-2.06, 1.56)	0/0.374 (-1.65, 7.32)	0/0.259 (-8.39, 2.55)	0/0.259 (-8.39, 2.55)	0/0.254 (-6.55, 1.96)	**1/0.032 (-6.57, -0.38)**	**1/0.032 (-6.57, -0.38)**
HYBRID(Original)	0/0.678 (-1.80, 8.51)	0/0.445185 (-4.32, 1.10)	0/0.781311 (-8.26, 8.62)	0/0.962872 (-4.72, 1.86)	0/0.351257 (-4.72, 1.86)	0/0.351257 (-6.68, 3.11)	0/0.430305 (-4.46, 2.68)	0/0.572715 (-4.46, 2.68)	0/0.572715 (-4.46, 2.68)
HYBRID(Two-level)	0/0.235 (-2.67, 6.33)	0/0.648 (-8.86, 3.53)	0/0.833 (-7.84, 9.50)	0/0.825 (-3.35, 4.10)	0/0.701 (-1.95, 1.37)	0/0.701 (-1.95, 1.37)	0/0.70 (-1.37, 1.95)	0/0.534 (-1.69, 0.95)	0/0.534 (-1.69, 0.95)
HYBRID(Three-level)	**1/0.015 (-9.27, -4.06)**	0/0.168 (-7.96, 1.62)	0/0.441 (-6.56, 3.11)	0/0.879 (-4.76, 4.14)	0/0.124 (-0.41, 2.87)	0/0.123 (-0.41, 2.87)	0/0.119 (-0.33, 2.43)	0/0.119 (-0.33, 2.43)	0/0.119 (-0.33, 2.43)
HYBRID (Four-level)	0/0.350 (-6.35, 1.17)	0/0.235 (-2.12, 7.59)	0/0.306 (-1.34, 3.79)	0/0.673 (-1.85, 2.74)	0/0.586 (-2.27, 1.36)	0/0.586 (-2.27, 1.36)	0/0.404 (-1.84, 0.82)	0/0.404 (-1.84, 0.82)	0/0.404 (-1.84, 0.82)

of accumulated failed links) which means when the failure process did not even start. These fluctuations are due to the latency metric for which a legacy system will take a longer time for routing reconfiguration than SDN-based counterpart as we have shown this result in our previous work [21]. Therefore, there are quite a few visible performance differences such as the two intervals in the middle of Fig. 7 [MESH(Three-level)]. Performance differences that occurred at $x = 0$ is not related to operational reliability.

Our findings in Figs. 6–8 show small differences in performance between legacy and SDN-based networks. Thus, there is a need to evaluate the statistical significance of these differences. Next, we employ a T-test [76] for this purpose.

C. Analyzing the Statistical Significance

In this subsection, we use the techniques proposed by Jain [76] to evaluate the statistical significance of the differences in reliability values as presented in Figs. 6–8.

There are 12 different topologies for each system as shown in the previous subsections. For each topology, there are θ observations per iteration of L. We take a random sample of 20 reliability values (i.e., $\theta = 20$) for both systems per each iteration of L. Let us represent the observations by $(l_0,\ l_1,\ l_2,\ldots..,l_{\theta-1})$ and $(s_0,\ s_1,\ s_2,\ldots..,s_{\theta-1})$ for each iteration applied on legacy and SDN-based systems, respectively. The differences of these observations are represented by $D = (s_0 - l_0,\ s_1 - l_1,\ s_2 - l_2,\ldots..,s_{\theta-1} - l_{\theta-1})$. Thus, the null and alternative hypotheses are as follows.

1) Null hypothesis H_0: $\mu_D = 0$. That is, the mean difference of D is equal to zero.
2) Alternative hypothesis H_a: $\mu_D \neq 0$.

The values of the reliability metric are roughly symmetric with no outliers. That is, the individual observations can be considered independent, the sampling distribution of mean difference is approximately normal, and the data are random samples from the population. Therefore, the conditions for performing this type of test are met. We used a significance level of $\alpha = 0.05$. We reject the null hypothesis if $p-$value $\leq \alpha$. Rejecting the null hypothesis means that the difference between the reliability performance for legacy and SDN-based networks is statistically *significant* which means that the test favors the alternative hypothesis. In this case, the zero value is not contained in the confidence interval (CI).

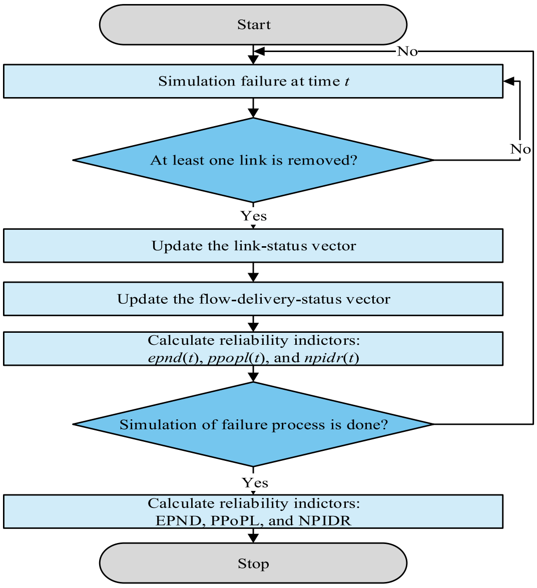
Start
Simulation failure at time t
At least one link is removed?
No
Yes
Update the link-status vector
Update the flow-delivery-status vector
Calculate reliability indictors:
$epnd(t)$, $ppopl(t)$, and $npidr(t)$
Simulation of failure process is done?
No
Yes
Calculate reliability indictors:
EPND, PPoPL, and NPIDR
Stop

Fig. 9. Flow chart of calculating reliability indicators through simulation.

TABLE V
COMPARISON OF TREE-TYPE NETWORK-RELIABILITY INDICATORS BETWEEN SDN-BASED VERSUS LEGACY TOPOLOGIES

Tree topology \ Reliability indicators		*EPND* (packet per time unit)	*PPoPL*	*NPIDR* (time%)
SDN-based topology	One-level	7.99	1	66.66
	Two-level	41.33	1	73.81
	Three-level	106.45	1	80.65
	Four-level	193.11	1	80.46
Legacy topology	One-level	7.79	1	64.99
	Two-level	41.42	1	73.96
	Three-level	104.77	1	79.37
	Four-level	192.11	1	80.05
The improvement of the SDN impact (%)	One-level	-2.56	0	-2.56
	Two-level	0.20	0	0.20
	Three-level	-1.60	0	-1.61
	Four-level	-0.52	0	-0.52

TABLE VI
COMPARISON OF MESH-TYPE NETWORK-RELIABILITY INDICATORS BETWEEN SDN-BASED VERSUS LEGACY TOPOLOGIES AND UNDER EXPONENTIAL FAILURE DISTRIBUTION

Mesh topology \ Reliability indicators		*EPND* (packet per time unit)	*PPoPL*	*NPIDR* (time%)
SDN-based topology	One-level	9.13	1	76.04
	Two-level	39	1	69.64
	Three-level	100.27	1	75.96
	Four-level	186.63	1	77.76
Legacy topology	One-level	8.99	1	74.99
	Two-level	40.18	1	71.75
	Three-level	103.41	1	78.34
	Four-level	181.38	1	75.57
The improvemen t of the SDN impact (%)	One-level	-1.39	0	-1.39
	Two-level	2.94	0	2.94
	Three-level	3.03	0	3.03
	Four-level	-2.89	0	-2.89

Table IV lists all results of the T-tests against the null hypothesis at a significance value of 0.05 and $\theta = 20$. There are 12 topologies and 9 iterations of operator L. We recall that the value 9 refers to the number of iterations of operator L, while 20 is the number of repetitions of each experiment. Therefore, for each iteration of L there will be 20 reliability values/observations.

Table IV shows that the majority of the CIs contain the value zero (98 cases out of 108 cases). This means that there is not enough evidence to reject the null hypothesis. The 10 cases that rejected the null hypothesis, highlighted in yellow in Table IV, do not establish a solid ground that can favor either system against the other as shown by their CIs. Therefore, it is plausible that there is no difference between the means of the performance of both systems concerning the reliability metric.

Table IV also shows that the insignificance of performance between both systems with respect to the reliability metric is more obvious in larger network types in terms of the number of nodes, links, and node degree. For example, unlike the tree network type, a hybrid type has higher interconnectivity which makes it more resistant to link failures. In other words, in larger networks, there is no single link failure that could disconnect the entire hybrid systems in early iterations of the failure process. This property allows a smooth degradation of reliability values in both systems and, hence, a more accurate evaluation of the mean difference of these values.

We further quantified the reliability difference between the two architecture designs by employing the reliability indicators $EPND$, $PPoPL$, and $NPIDR$ that we introduced in Section IV-B.

D. Reliability Indicators

Reliability indicators $EPND$, $PPoPL$, and $NPIDR$ will be used in this subsection for deeper quantitative insights into reliability. This will shed light to better interpret the results at packet-level delivery. Fig. 9 shows the flow chart of calculating these indicators through simulation. The flowchart, in Fig. 9, starts with a link-failure at time t (i.e., apply L each time). Next, if at least one link is removed, it updates the link-state vectors so that the states of removed links are set to zero, and then updates the flow-delivery-status vector. In our experiments, the flow-delivery-status vector includes $n_k = |H|(|H| - 1)$ values that represent all-to-all communications of the pingAll() function, where $|H|$ is the number of hosts in each network topology. In the next step, reliability indicators at time t are calculated according to (6)–(8). If the failure process is done, the network reliability indicators for all time t (i.e., $EPND$, $PPoPL$, and $NPIDR$) are calculated. Otherwise another failure iteration takes place.

We use the data collected in the experiments described in Section V(B) to calculate the results of the reliability indicators. The results are provided in Tables V– VII for the tree, mesh, and hybrid topologies, respectively. PPoPL is always one in both architecture designs because the link failure is present in every iteration. The cross-reference comparison between Table V and Fig. 6 shows the same general behavior of the performance of the two network designs. However, values of EPND and NPIDR provide detailed information. For instance, SDN-related network

TABLE VII
COMPARISON OF HYBRID-TYPE NETWORK-RELIABILITY INDICATORS BETWEEN SDN-BASED VERSUS LEGACY TOPOLOGIES AND UNDER EXPONENTIAL FAILURE DISTRIBUTION

Reliability indicators / Hybrid topology		$EPND$ (packet per time unit)	$PPoPL$	$NPIDR$ (time%)
SDN-based topology	One-level	37.61	1	67.16
	Two-level	188.72	1	78.64
	Three-level	448.09	1	81.18
	Four-level	879.42	1	88.65
Legacy topology	One-level	36.56	1	65.28
	Two-level	191.18	1	79.66
	Three-level	438.55	1	79.45
	Four-level	887.26	1	89.44
The improvement of the SDN impact (%)	One-level	-2.89	0	-2.89
	Two-level	1.28	0	1.28
	Three-level	-2.18	0	-2.18
	Four-level	0.88	0	0.88

does not make any improvement over the legacy counterpart in one-level, three-level, and four-level topologies.

On the contrary, legacy networks performed slightly better under these networks with a high 2.5% at one-level topology. For two-level topology, SDN made a tiny improvement with only 0.2% over the legacy network. Note that the improvement values of the EPND and their corresponding NPIDR should be equal as per (8).

Table VI shows that SDN has improvement in two-level and three-level mesh topologies, respectively, with 2.94% and 3.03% improvement. This finding confirms the results in Fig. 7. On the other hand, the legacy network did better in one-level and four-level topologies, with 1.39% and 2.89%, respectively.

Table VII shows the performance of SDN-related networks, and their legacy counterpart is almost even. Legacy and SDN have performance peaks of 2.89% and 1.28%, respectively. Regardless of the underlying network architecture design, the results from all the conducted experiments above showed a comparable performance of both traditional and SDN-based systems.

Furthermore, the failure process had a greater impact on the reliability of smaller topologies such as tree topology and its extensions. This is probably due to the direct effect of the link failure process on the cut-edge sets with the smallest size as defined in (3).

E. Sensitivity Analysis to Hazard Rate

In our previous experiments, the exponential distribution was adopted with a constant hazard rate (i.e., λ is constant) for all links. However, the hazard rate could be different from one link to another. This subsection addresses the sensitivity of the reliability metric to the hazard rate. Therefore, we will allow each link in a network to have a different hazard rate rather than a fixed one. We choose the Weibull distribution for this purpose. Unlike exponential distribution, Weibull distribution allows time-dependent failure rates [77]. For example, Weibull distribution considers the network elements' early-operational life (infant mortality), then fairly constant failure rate (useful life), and the aging of network elements' in which failure rate increases again (wear out). Fig. 10 shows the hazard rates of the three stages of an element's lifetime that is considered by Weibull distribution.

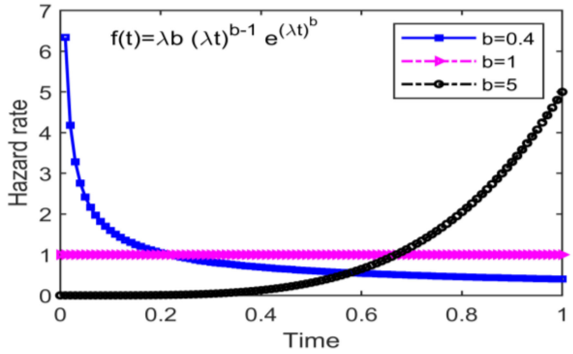

Fig. 10. Bathtub curve obtained by multiple failure rates modeled with Weibull distribution using $\lambda = 1$ and different values of b.

TABLE VIII
COMPARISON OF EPND INDICATOR BETWEEN SDN-BASED VERSUS LEGACY TOPOLOGIES AND UNDER WEIBULL DISTRIBUTION WITH $\lambda = 1$ AND $b = 0.4$

Reliability indicators / Hybrid topology		$EPND$ $(TREE)$	$EPND$ $(MESH)$	$EPND$ $(HYBRID)$
SDN-based topology	One-level	8.4	8.83	42
	Two-level	41.94	37.61	186.5
	Three-level	108.22	97.36	386.69
	Four-level	197	159.62	708
Legacy topology	One-level	8.4	8.94	42.22
	Two-level	41.83	37.44	186.45
	Three-level	107.89	97.64	386.19
	Four-level	196.75	162.27	708.73
The improvement of the SDN impact (%)	One-level	0	1.24	0.53
	Two-level	-0.27	-0.45	-0.02
	Three-level	-0.31	0.28	-0.13
	Four-level	-0.13	1.64	0.10

Our main focus in this subsection is to assess the impact, if any, of different hazard rates of network links on the performance evaluation of both architecture designs against the reliability metric. To this end, we evaluate the performance through either hazard rates at the infant mortality stage or at the wear-out stage. In our experiment, we chose the Weibull distribution with $b = 0.4$ and $\lambda = 1$ (i.e., infant mortality-based hazard rates). The results are shown in Table VIII.

The findings of this experiment show no impact of different hazard rates on the comparative performance evaluations of legacy and SDN-based network against network reliability metric. That is, the reliability of both network architecture designs is not sensitive to different hazard rates.

F. Discussion

Our simulation results show that the performance of the two architecture designs (i.e., SDN and legacy) have comparable performance regarding operational reliability and the difference between them is deemed statistically insignificant. Section V-C sheds light on packet-level details of this comparability in the performance. Tables V–VII reassure the comparable performance for both systems with respect to EPND/NPIDR values. Furthermore, the values of EPND in both designs increase as the topology becomes deeper (have more levels). This behavior is likely due to the fact that a deeper topology has more traffic flows, and hence a higher probability of losing more packets upon link failure.

The main reason that SDN does not improve the operational reliability is that the OpenFlow-driven approach suffers from a convergence problem: The control-plane would take too much response time to detect the failure, and then send updates to be installed in the impacted OVSwitches. This response time is not fast enough to handle some application parameters such as TIMEOUT interval (e.g., the ping tool has 4 s as a default timeout). Meanwhile, a large number of packets will be dropped, and new packets could be retransmitted (e.g., the TCP/IP packets), which generates an overhead in network loading and causes routing interruption.

It would be wise for the network operators to migrate their legacy network to an SDN-based network to gain from the wide range of benefits of SDN even though the reliability does not show improvement in our study. Reliability enhancement in the SDN-based network would still be possible after the migration. Next, we suggest some proposals for reliability improvement in SDN-based network.

1) Cut-edge set has a significant impact on network reliability as we showed in our analysis. The smaller the set, the higher probability to defragment prematurely and hence the sharper decline in the reliability. The minimum cut-edge set is related to several graph-theoretic problems such as the Max-flow Min-cut theorem [78] to maximize the flow/throughput in a network, and Menger's theorem to find the number of edge-disjoint paths. Therefore, it is a good design practice to avoid smaller cut sets in a network. In fact, maximizing the minimum cut-edge set should be optimized along with constraints such as cost and complexity.
2) Upon a link failure, path computation element protocol (PCEP) is used to compute the shortest alternative paths that reconnect the source and destination nodes. It then will pick any path of the potential equal-cost multipath (ECMP). However, the chosen path may not be the best fit from the reliability point of view. Therefore, this tie among the ECMP paths should be broken in a way that enhances network reliability. To this end, path disjointness provides a good criterion to break this tie because it is aligned with the traffic engineering (TE) concept that seeks to optimize the network performance, including network reliability. Our analysis suggested computing an alternative path that does not share links with the primary path (i.e., the path selected prior to the failure) to ensure that a failed link will not affect the backup paths. That is, a link-disjoint-enabled ECMP.
3) Our findings pointed to a convergence problem in the SDN-based network upon link failure. This problem can be mitigated by avoiding a large amount of unnecessary data traffic that is trying to stream through a failed link; a solution could be to locally inform the relevant switches, perhaps the ones within one hop from the failed link, to stop sending packets through the failed links. In this case, a fast reaction will be taken locally from the neighboring switches and even before the SDN controller responded by updating the routing entries. Hence, improving operational reliability.

The limitations of our study include various failures of network elements. We did not address the reliability under nodal failure, which could affect larger parts of the network. The deployment of multiple controllers is another limitation that could perhaps reduce the overhead communication on a single controller and, hence, enhance the reliability. Although our assumption to not restore a failed link is valid from the probability evaluation's point of view, it limited our insight to the criticality of network links (i.e., the degradation in reliability associated with a specific link removal independent from the failure of others).

VI. Conclusion

This article provides an insight into network reliability under link failure scenarios and conducts comparative performance analysis between two architecture designs of communication networks: Legacy network and SDN-based network. Experimental results demonstrated that SDN-based networks have comparable performance to legacy networks against operational reliability. We validate these findings by using the T-test as a statistical tool. We anticipate that this study will also be of great importance to mobile network operators who migrated/to migrate their networks from legacy to SDN. Our analysis and findings will assist the operators in reviewing and evaluating their migration to SDN from a network reliability perspective.

The results of our experiments suggested that fast failure detection and recovery was crucial to enhance operational reliability in SDN-based networks. Furthermore, the network designers/planners should pay special attention to the reachability metric and various failure drivers in SDN, such as timeout parameter, unreachability, and other hardware/software failures.

As part of future work, we plan to extend this study to consider targeted failures according to some centrality measures such as betweenness centrality under dependent link failures and using real data traffic. As a network (or part of it) could belong to multi operators/owners and, hence, have various link failure distributions, an interesting problem to investigate would be how cooperative networks would behave upon link failure? and to what extent they would still provide reliable services under some routing constraints.

References

[1] S. Song, H. Park, B.-Y. Y. Choi, T. Choi, and H. Zhu, "Control path management framework for enhancing software-defined network (SDN) reliability," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 2, pp. 302–316, Jun. 2017.

[2] "Cisco visual networking index: Forecast and methodology, 2016–2021 - Cisco," Accessed: Mar. 28, 2020, [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral /service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[3] Y. Al Mtawa, A. Haque, and B. Bitar, "Does Internet of things disrupt residential bandwidth consumption?," in *Proc. IEEE 88th Veh. Technol. Conf.*, Dec. 2018, pp. 1–6.

[4] Y. Al Mtawa, A. Haque, and B. Bitar, "The mammoth Internet: Are we ready?," *IEEE Access*, vol. 7, pp. 132894–132908, 2019.

[5] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 107–115, Feb. 2015.

[6] S. Gorlatch, T. Humernbrum, and F. Glinka, "Improving QoS in real-time internet applications: From best-effort to software-defined networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 189–193.

[7] H. Thorisson, J. H. Lambert, J. J. Cardenas, and I. Linkov, "Resilience analytics with application to power grid of a developing region," *Risk Anal*, vol. 37, no. 7, pp. 1268–1286, Jul. 2017.

[8] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[9] Z. Yang and K. L. Yeung, "SDN candidate selection in hybrid IP/SDN networks for single link failure protection," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 312–321, Feb. 2020.

[10] I. Gertsbakh and Y. Shpungin, "Lomonosov's turnip," in *Network Reliability.*, Singapore: Springer, 2020, pp. 71–80.

[11] L. Cominardi, C. J. Bernardos, P. Serrano, A. Banchs, and A. de la Oliva, "Experimental evaluation of SDN-based service provisioning in mobile networks," *Comput. Stand. Interfaces*, vol. 58, pp. 158–166, May 2018.

[12] Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3220–3258, May 2018.

[13] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, "SDN based mobile networks: Concepts and benefits," *Wireless Pers. Commun.*, vol. 78, no. 3, pp. 1629–1644, Oct. 2014.

[14] A. Tzanakaki *et al.*, "Converged wireless access/optical metro networks in support of cloud and mobile cloud services deploying SDN principles," in *Fiber-Wireless Convergence in Next-Generation Communication Networks*. Cham, Switzerland: Springer, 2017, pp. 359–388.

[15] M. Karakus and A. Durresi, "Economic analysis of software defined networking (SDN) under various network failure scenarios," in *Proc. IEEE Int. Conf. Commun.*, May 2019, pp. 1–6.

[16] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3259–3306, Oct. 2018.

[17] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, "A survey of deployment solutions and optimization strategies for hybrid SDN networks," *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, pp. 1483–1507, Apr. 2019.

[18] K. Benzekki, A. El Fergougui, and A. E. Elalaoui, "Software-defined networking (SDN): A survey," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5803–5833, Dec. 2016.

[19] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *J. Netw. Comput. Appl.*, vol. 80, no. Supplement C, pp. 200–218, Feb. 2017.

[20] A. Shirmarz and A. Ghaffari, "Performance issues and solutions in SDN-based data center: A survey," *J. Supercomput.*, pp. 1–49, Jan. 2020, doi: 10.1007/s11227-020-03180-7.

[21] Y. Al Mtawa, A. Memari, A. Haque, and H. Lutfiyya, "Evaluating QoS in SDN-Based EPC: A comparative analysis," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf.*, Jun. 2019, pp. 1279–1286.

[22] S. Saraswat, V. Agarwal, H. P. Gupta, R. Mishra, A. Gupta, and T. Dutta, "Challenges and solutions in software defined networking: A survey," *J. Netw. Comput. Appl.*, vol. 141, pp. 23–58, Sep. 2019.

[23] M. Desai and T. Nandagopal, "Coping with link failures in centralized control plane architectures," in *Proc. 2nd Int. Conf. Commun. Syst. Netw.*, 2010, pp. 1–10.

[24] X. Zhang, Z. Cheng, R. P. Lin, L. He, S. Yu, and H. Luo, "Local fast reroute with flow aggregation in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 785–788, Apr. 2017.

[25] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Comput. Netw.*, vol. 83, pp. 280–296, Jun. 2015.

[26] E. Sakic and W. Kellerer, "Response time and availability study of RAFT consensus in distributed SDN control plane," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 304–318, Mar. 2018.

[27] S. Wu, L. Yang, J. Guo, Q. Chen, X. Liu, and C. Fan, "Intelligent quality of service routing in software-defined satellite networking," *IEEE Access*, vol. 7, pp. 155281–155298, 2019.

[28] F. J. Ros and P. M. Ruiz, "Five nines of southbound reliability in software-defined networks," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 31–36.

[29] S. Guner, G. Gur, and F. Alagoz, "Proactive controller assignment schemes in SDN for fast recovery," in *Proc. Int. Conf. Inf. Netw.*, Mar. 2020, pp. 136–141.

[30] J. Xie, D. Guo, X. Zhu, B. Ren, and H. Chen, "Minimal fault-tolerant coverage of controllers in IaaS datacenters," *IEEE Trans. Services Comput.*, vol. 13, no. 6, pp. 1128–1141, Nov./Dec. 2020.

[31] P. Wang, H. Xu, L. Huang, C. Qian, S. Wang, and Y. Sun, "Minimizing controller response time through flow redirecting in SDNs," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 562–575, Feb. 2018.

[32] D. Santos, A. De Sousa, and C. M. Machuca, "Combined control and data plane robustness of SDN networks against malicious node attacks," in *Proc. 14th Int. Conf. Netw. Service Manage.*, 2018, pp. 54–62.

[33] G. Nencioni, B. E. Helvik, and P. E. Heegaard, "Including failure correlation in availability modeling of a software-defined backbone network," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 1032–1045.

[34] M. Nishino, T. Inoue, N. Yasuda, S.-I. Minato, and M. Nagata, "Optimizing network reliability via best-first search over decision diagrams," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1817–1825.

[35] W.-C. Yeh, "An improved sum-of-disjoint-products technique for symbolic multi-state flow network reliability," *IEEE Trans. Reliab.*, vol. 64, no. 4, pp. 1185–1193, Dec. 2015.

[36] Y. Mo, L. Xing, F. Zhong, Z. Pan, and Z. Chen, "Choosing a heuristic and root node for edge ordering in BDD-based network reliability analysis," *Reliab. Eng. Syst. Saf.*, vol. 131, pp. 83–93, Nov. 2014.

[37] G. Hardy, C. Lucet, and N. Limnios, "K-Terminal network reliability measures with binary decision diagrams," *IEEE Trans. Reliab.*, vol. 56, no. 3, pp. 506–515, Sep. 2007.

[38] S.-Y. Kuo, F.-M. Yeh, and H.-Y. Lin, "Efficient and exact reliability evaluation for networks with imperfect vertices," *IEEE Trans. Reliab.*, vol. 56, no. 2, pp. 288–300, Jun. 2007.

[39] L. Xing, "An efficient binary-decision-diagram-based approach for network reliability and sensitivity analysis," *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Humans*, vol. 38, no. 1, pp. 105–115, Jan. 2008.

[40] Y. Niu and F.-M. Shao, "A practical bounding algorithm for computing two-terminal reliability based on decomposition technique," *Comput. Math. with Appl.*, vol. 61, no. 8, pp. 2241–2246, Apr. 2011.

[41] J. Carlier and C. Lucet, "A decomposition algorithm for network reliability evaluation," *Discrete Appl. Math.*, vol. 65, no. 1–3, pp. 141–156, Mar. 1996.

[42] M.-L. Rebaiaia and D. Ait-Kadi, "System reliability evaluation for imperfect networks using polygon-to-chain reduction," *Amer. J. Oper. Res.*, vol. 07, no. 03, pp. 201–224, May 2017.

[43] R. K. Wood, "A factoring algorithm using polygon-to-chain reductions for computing K-terminal network reliability," *Networks*, vol. 15, no. 2, pp. 173–190, 1985.

[44] M. Manzano, F. Sahneh, C. Scoglio, E. Calle, and J. L. Marzo, "Robustness surfaces of complex networks," *Sci. Rep.*, vol. 4, no. 1, pp. 1–6, Sep. 2014.

[45] K. Han, T. A. Nguyen, D. Min, and E. M. Choi, "An Evaluation of Availability, Reliability and Power Consumption For a SDN Infrastructure Using Stochastic Reward Net," in *Advances in Computer Science and Ubiquitous Computing.* Singapore: Springer, pp. 637–648, 2016.

[46] J. Ortiz and D. Culler, "Multichannel reliability assessment in real world WSNs," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 162–173.

[47] A. Elsayed, *Reliability Engineering*. Wiley, Hoboken, NJ, USA, 2013.

[48] A. Sadasivarao, D. Naik, C. Liou, S. Syed, and A. Sharma, "Demystifying SDN for optical transport networks: Real-world deployments and insights," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–7.

[49] H. Cetinay, C. Mas-Machuca, J. L. Marzo, R. Kooij, and P. Van Mieghem, "Comparing destructive strategies for attacking networks," in *Guide to Disaster-Resilient Communication Networks*. J. Rak and D. Hutchison, Eds., Cham, Switzerland: Springer, 2020, pp. 117–140.

[50] W. K. Ghamry and K. M. F. Elsayed, "Network design methods for mitigation of intentional attacks in scale-free networks," *Telecommun. Syst.*, vol. 49, no. 3, pp. 313–327, Mar. 2012.

[51] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "Network vulnerability to single, multiple, and probabilistic physical attacks," in *Proc. - IEEE Mil. Commun. Conf.*, 2010, pp. 1824–1829.

[52] M. Rahnamay-Naeini, J. E. Pezoa, G. Azar, N. Ghani, and M. M. Hayat, "Modeling stochastic correlated failures and their effects on network reliability," in *Proc. - Int. Conf. Comput. Commun. Netw.*, 2011, pp. 1–6.

[53] S.-Y. Wang, L.-M. Chen, S.-K. Lin, and L.-C. Tseng, "Using SDN congestion controls to ensure zero packet loss in storage area networks," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage.*, May 2017, pp. 490–496.

[54] D. Sidhu, R. Nair, and S. Abdallah, "Finding disjoint paths in networks," in *Proc. Conf. Commun. Architecture Protocols*, 1991, pp. 43–51.

[55] Y. Egawa, A. Kaneko, and M. Matsumoto, "A mixed version of Menger's theorem," *Combinatorica*, vol. 11, no. 1, pp. 71–74, Mar. 1991.

[56] "Mininet: Network emulator/simulator," Accessed: Mar. 20, 2020. [Online]. Available: http://mininet.org/

[57] "Using openflow — Open vSwitch 2.9.90 documentation," Accessed Mar. 30, 2020, [Online]. Available: http://docs.openvswitch.org/en/latest/faq/openflow/

[58] A. J. Pinheiro, E. B. Gondim, and D. R. Campelo, "An efficient architecture for dynamic middlebox policy enforcement in SDN networks," *Comput. Netw.*, vol. 122, pp. 153–162, Jul. 2017.

[59] R. Barrett, A. Facey, W. Nxumalo, J. Rogers, P. Vatcher, and M. St-Hilaire, "Dynamic traffic diversion in SDN: Testbed vs mininet," in *Proc. Int. Conf. Comput., Netw. Commun.*, Mar. 2017, pp. 167–171.

[60] J. Wang, S. Zhang, W. Chen, D. Kong, X. Zuo, and Z. Yu, "Design and implementation of SDN-Based underwater acoustic sensor networks with multi-controllers," *IEEE Access*, vol. 6, pp. 25698–25714, May 2018.

[61] D. Kreutz, J. Yu, F. M. V. Ramos, and P. Esteves-Verissimo, "Anchor: Logically centralized security for software-defined networks," *ACM Trans. Privacy Secur.*, vol. 22, no. 2, Feb. 2019, Art. no. 8.

[62] H. Ghalwash and C. H. Huang, "A QoS framework for SDN-Based networks," in *Proc. 4th IEEE Int. Conf. Collaboration Internet Comput.*, Nov. 2018, pp. 98–105.

[63] D. Adami, L. Donatini, S. Giordano, and M. Pagano, "A network control application enabling software-defined quality of service," in *Proc. IEEE Int. Conf. Commun.*, Sep. 2015, pp. 6074–6079.

[64] D. Muelas, J. Ramos, and J. E. L. De Vergara, "Assessing the limits of mininet-based environments for network experimentation," *IEEE Netw.*, vol. 32, no. 6, pp. 168–176, Nov. 2018.

[65] J. Castillo-Lema, A. Venancio Neto, F. De Oliveira, and S. Takeo Kofuji, "Mininet-NFV: Evolving Mininet with OASIS TOSCA NVF profiles towards reproducible NFV prototyping," in *Proc. IEEE Conf. Netw. Softwarization: Unleashing Power Netw.*, Jun. 2019, pp. 506–512.

[66] F. H. M. B. Lima, L. F. M. Vieira, M. A. M. Vieira, A. B. Vieira, and J. A. M. Nacif, "Water ping: ICMP for the internet of underwater things," *Comput. Netw.*, vol. 152, pp. 54–63, Apr. 2019.

[67] Floodlight, "Floodlight openflow controller," 2014. Accessed Mar. 20, 2020, [Online]. Available: http://www.projectfloodlight.org/floodlight/

[68] L. Cheng and S.-Y. Wang, "Application-aware SDN routing for big data networking," in *Proc. IEEE Global Commun. Conf.*, 2015, pp. 1–6.

[69] S. Chakraborty, N. K. Goyal, S. Mahapatra, and S. Soh, "Minimal path-based reliability model for wireless sensor networks with multistate nodes," *IEEE Trans. Reliab.*, vol. 69, no. 1, pp. 382–400, Mar. 2020.

[70] H. Cancela, L. Murray, and G. Rubino, "Efficient estimation of stochastic flow network reliability," *IEEE Trans. Reliab.*, vol. 68, no. 3, pp. 954–970, May 2019.

[71] D. H. Huang, C. F. Huang, and Y. K. Lin, "A binding algorithm of lower boundary points generation for network reliability evaluation," *IEEE Trans. Reliab.*, vol. 69, no. 3, pp. 1087–1096, Sep. 2020.

[72] H. M. F. AboElFotoh, S. S. Iyengar, and K. Chakrabarty, "Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures," *IEEE Trans. Reliab.*, vol. 54, no. 1, pp. 145–155, Mar. 2005.

[73] S. Xiang and J. Yang, "K-Terminal reliability of ad hoc networks considering the impacts of node failures and interference," *IEEE Trans. Reliab.*, vol. 69, no. 2, pp. 725–739, Jun. 2020.

[74] A. Heidarzadeh, A. Sprintson, and C. Singh, "A fast and accurate failure frequency approximation for k-terminal reliability systems," *IEEE Trans. Reliab.*, vol. 67, no. 3, pp. 933–950, Sep. 2018.

[75] S. Jiang, D. He, and J. Rao, "A prediction-based link availability estimation for mobile ad hoc networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2001, vol. 3, pp. 1745–1752.

[76] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, NY, USA: Wiley, 1990.

[77] H. Rinne, *The Weibull Distribution*. Baco Raton, FL, USA: Chapman and Hall/CRC, 2008.

[78] G. B. Dantzig and D. R. Fulkerson, "On the max flow min cut theorem of networks," in *Linear Inequalities Related Systems*, vol. 38, Princeton, NJ, USA: Princeton Univ. Press, 1955, pp. 215–221.

Yaser Al Mtawa received the Ph.D. degree in computer science from Queen's University, Kingston, ON, Canada, in 2017.

From 2012 to 2016, he was a Research Assistant with the School of Computing, Queen's University. Since 2017, he has been a Postdoctoral Research Fellow with the Computer Science Department, Western University, London, Canada. He is currently involved in multiple projects in collaboration with giant industry partners such as Bell Canada, TELUS, IBM Canada, and Juniper Networks. His research interest includes the Internet of Things (IoT) with a particular focus on wireless sensor networks (WSNs) and smart homes and smart power grids, graph-theoretic network problems, network reliability, and software-defined networking (SDN).

Dr. Al Mtawa was the recipient of the Ontario Graduate Scholarship (OGS), the Robert Sutherland Fellowship, SOSCIP TalentEdge Fellowship, Mitacs Accelerate Fellowship, and the Kuwait Emir's Golden Medal Award for the highest academic standing.

Anwar Haque received the Ph.D. degree in electrical and computer engineering and the M.Math degree in computer science from the University of Waterloo, Waterloo, ON, Canada, 2013 and 2017, respectively.

He is an Assistant Professor with the Department of Computer Science, University of Western Ontario, London, ON, Canada. He also serves as Industry Expert in Residence with the faculty of Science, University of Western Ontario. Before joining the University of Western Ontario, he was an Associate Director with Bell Canada. Dr. Haque is the director of the WING Lab, where he conducts cutting-edge research in emerging network technologies and smart services.. His research interests include wireless/wireline network resources and performance management/optimization and cyber-security, focusing on IoT-based smart services and applications.

Dr. Haque was the recipient of the IEEE ISNCC 2020 best paper award, IEEE CCECE 2020 leadership award, and several national/provincial-level research grants, including NSERC, MITACS, and OCE. He is currently an Associate Editor for the *Elsevier Vehicle Communications Journal* and the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING.

Hanan Lutfiyya received the Ph.D. degree from the Missouri University of Science and Technology, Rolla, Missouri. She is a Professor and Chair of the Department of Computer Science at the University of Western Ontario, London, ON, USA. Her research group in collaboration with industrial and government partners investigates different aspects of reliable software and systems. She is currently collaborating with Tillsonburg Hydro on smart grids. Her research interests include Internet of Things, software engineering, self-adaptive systems, autonomic computing, monitoring and diagnostics, mobile systems, policies, and clouds.

Professor Lutfiyya was the recipient of funding from Ontario Research Fund (ORF), NSERC, IBM, Samsung, Fujitsu and Canada's Communications Research Centre (CRC). She was also the recipient of the UWO Faculty Scholar Award, in 2006. She is currently an Associate Editor for the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, and has recently served as Program Co-Chair of IEEE/IFIP Network Operations and Management Symposium and the IEEE International Conference on Network and Service Management (CNSM).