International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 13, No.1 (Jan-2023)

http://dx.doi.org/10.12785/ijcds/130117

# HSPC-SDN: Heuristic Driven Self-Configuring Proactive Controller for QoS-Centric Software Defined Network

**S Sharathkumar[1] and N Sreenath[2]**

[1]*Department of CSE, Puducherry Technological University, Puducherry, India*
[2]*Department of CSE, Puducherry Technological University, Puducherry, India*

*Received 23 Jan. 2022, Revised 8 Jul. 2022, Accepted 12 Jan. 2023, Published 31 Jan. 2023*

**Abstract:**The exponential rise in software computing, low-cost hardware and allied application demands has broadened the horizon for wireless technologies to serve different purposes. Wireless communication systems being central to the modern innovation and industrial growth have given rise to the different communication ecosystems including internet of things, machine to machine communication, wireless local area network, Ad-hoc networks etc. However, coping with non-negotiable service level agreements have forced industries to ensure quality of service (QoS) and quality of experience demands. To meet such demands, software defined network (SDN) has gained widespread attention. The ability to enable higher programmability, flexibility and scalability makes SDN-based system viable; yet, guaranteeing their robustness towards dynamic network, link-failure and adaptive QoS-centric recovery has remained a challenge. In sync with this motive, in this paper a robust Heuristic Driven Self-Configuring Proactive Controller is designed for QoS-centric SDN network (HSPC-SDN). Unlike classical data-plane SDN controllers or allied routing solutions, HSPC-SDN performs multi-constraints risk assessment followed by heuristic driven disjoint multiple path selection to support proactive network failure-recovery. HSPC-SDN applies dynamic link-quality information, cumulative congestion degree, probability of successful transmission and link quality change index to perform best forwarding device selection to alleviate any malicious behaviour or malfunction during transmission. Subsequently, it applies genetic algorithm to perform disjoint multiple forwarding cum failure recovery path selection that in conjunction with AND logic function enables self-configuring route recovery to meet fault-tolerant QoS-centric communication. The proposed heuristic model exploits network availability information amalgamated with minimal distance and strictly no-shared component criteria to perform multiple disjoint forwarding-path cum recovery-path selection. Simulation based results revealed that HSPC-SDN, which can be implemented as a standalone single data-plane controller as well as a middleware routing concept achieves superior average packet delivery rate of 98.03%, packet loss rate of 1.97%, recovery time of 1.66ms and energy consumption of 77.14mJ over other disjoint forwarding path based SDN controllers.

**Keywords:** Software Defined Network, Fault-Tolerant Communication, Quality-of-Service, Self-configuring Network

## 1. INTRODUCTION

Communication systems being the vital component of next-generation technologies have gained widespread attention of academia-industries to achieve reliable and quality-of-service (QoS) oriented service provision.However, the diversity of applications and operating environment make real-time realization a challenging task. Different technologies including wireless networks,internet enabled networks etc. serving significantly large purposes have gained inevitable role across industrial horizon. Yet, guaranteeing the reliable services irrespective of the operating conditions and network dynamism has remained challenge for industries. On the other hand, rising market competition too has forced stakeholders to achieve more efficient, consistent and reliable network solution to serve users. Despite the fact that the next-generation technologies like internet of things (IoT) and machine-to-machine (M2M) have emerged as the most sought-after technologies to fulfill major communication demands; yet network dynamism and resulting link-failure over dynamic operating condition has remained a challenge [1] [2]. In sync with such problems, alleviating the likelihood of network failure is inevitable to fulfill committed QoS [3] [4].Specially, fulfilling QoS turns out to be inevitable in business running under the umbrella of strict service level agreement (SLA). To cope up with the QoS or quality of experience (QoE) demands and probable failure likelihood, strengthening network solution is the only viable solution or remedies. The viable remedies often target on avoiding any disruption in network-level QoS and application-level QoE to the maximum possible extent. Noticeably, in major communication systems failure might take place due to hardware failure, node death, link-disruption, flooding, and link-outage or physical network damages etc. To avoid any detriment to the intended QoS

*E-mail address: sharathkumar.s@pec.edu,nsreenath@ptuniv.edu.in*

or QoE provision, fault-tolerance is identified as the best measure. However, achieving these objectives require network to be more programmable, flexible and innovative. To met these demands, software defined networking (SDN) has emerged as a vital technology [5]–[8]. The ability to have the central visualization and control enables SDN to become the mainstream paradigm to serve fault-tolerant or fault-resilient networks solution. SDN-based networks possess high network visibility along with decoupled data and control plane functionality that make it more fault-resilient [3] [5]–[8].As depicted in Fig. 1, SDN comprises numerous layers and planes; however, data plane, control plane and application layer are the more discussed one. Typically,the data plane layer, which is also called as the forwarding layer is responsible for handling data packets transmitted by the user(s) by means of the deployed forwarding nodes or the network devices. More specifically, it houses the forwarding network table and medium access control (MAC) as routers and switches to complete data transmission across the network. In fact,the data plane functions as a forwarding element where its functional behave is decided by the controller and hence often referred as a forwarding plane [9]. Unlike data plane, the control plane acts as a decision layer which decides the optimal way through which the data has to be forwarded across the deployed network.Typically, control plane is hypothesized to be the controller acting as a network brain to control overall communication [5]–[8]. Moreover, it functions as an intermediate layer in between the data plane and the application plane. In application plane, different network applications are employed to control the network-logic operating onto the top of the controller. The communication in between the control plane and application plane is feasible by means of the northbound API's like the REST API's [10]. Similarly, the communication in between the control plane and the data plane is accomplished by means of the southbound API's like OpenFlow protocol [7] [11]. The key motive of control plane is to update and synchronize the network table, while the application plane layer is accountable towards network applications and services. The decoupling of control and data plane enables transfer of the control logic's to the controller that helps network to gain superior fault-resilience. In SDP technology, controller enables retrieving the global view of the entire network and acts as a network brain to make proactive network decision [3]–[6]. Since, SDN controller possesses the ability to configure, re-configure and self-configure the forwarding devices based on certain predefined custom routing policies, it helps improving reliability even under dynamic operating conditions [10] [11]. It broadens the horizon for real-time purposes serving ad-hoc communications, IoT/M2M communication, vehicular communication, etc. [1] [2].Retaining aforesaid QoS-centric (dynamic) configuration requires a controller often called SDN-controller. A network can be designed with single SDN-controller as well as multiple controllers; however, single controller with a superiorly designed (programmable) routing can be more efficient towards resource constrained and delay resilient communication [12]. Despite efficacy,

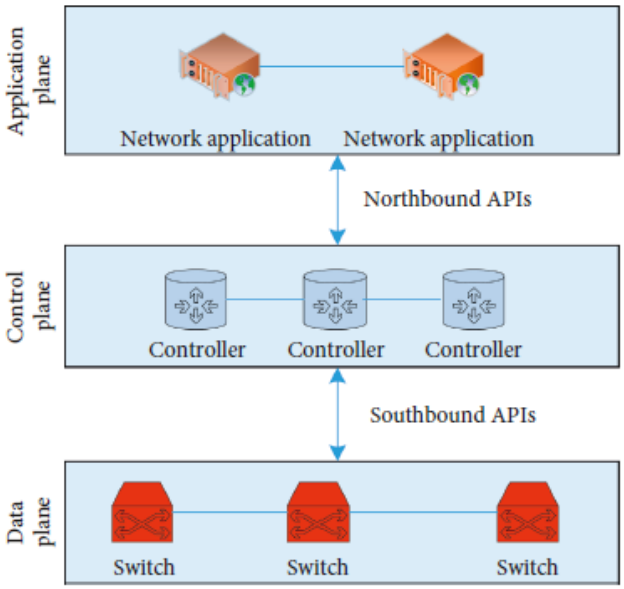


Figure 1. An Illustration of SDN Architecture

single controller-based network(s) might undergo single point of failure (SPOF) thus compromising the network performance. SPOF can also collapse the entire network and allied communication and hence can violate QoS/QoE or SLA agreements [13] [14]. To meet QoSdemands, especially in terms of consistency, availability and reliability, guaranteeing fault-tolerance is must for which developing a fault-resilient or fault-tolerant SDNcontroller is inevitable [12].

In the past a few efforts have been made towards fault-tolerance in SDN, mainly designed in two distinct domains; data plane and control plane approaches. Approaches designed for data-plane [10] focused on achieving the path failure recovery within 50 milliseconds. Similarly, researchers found that enabling fast-fail over model to react switch or link failure can be vital towards SDN. Interestingly, literature's identify that in classical SDN model(s), often undergo single-point of failure (SPOF), thus demanding a robust fault-tolerant SDN controller solution. In fact, SDN advocates applying a logically centralized controller to make suitable traffic forwarding or routing decision, where the deployed local switches can be employed to forward the packets in the data plane. Thus, it decouples control plane from physical data plane while retaining fault-tolerant transmission. These characteristics enable SDN to be employed in numerous purposes including test-beds, production networks, vehicular communication, indoor networks, inter-continental wireless area network (IC-WAN) [15] etc. Despite the fact that a number of efforts have been made towards SDN designs and deployment; yet very less attention has been made for the challenge called "network-failure" that turns out to be more severe in case of dynamic networks. Noticeably, network failure signifies link-outage

which can obstruct normal traffic-flow despite availability of the alternate path due to high latency. To alleviate this problem, authors [16] suggested failure-recovery concept; yet it doesn't guarantee QoS under probable iterative failure likelihood or malicious attacks [17]. Literature's assessment reveal that SDN data-plane approaches are more opt towards fault-tolerance [10]. These methods are broadly classified as reactive and proactive methods where the first requires relying on sophisticated controller, while the later may not require completely depending on any controller. In such case, proactive data-plane approach seems to be more effective towards SDN [18].

At the other hand fault-tolerant controller solutions are quite confined, especially in terms of their inability to cope up with probable risk assessment and proactive decision making, availability and consistency. It severely impacts overall QoS/QoE performance of SDN networks [19] [20]. Well-known approaches like OpenFlow [7] manages communication in between the controllers and switches [21]; however, so far it could not address any malicious assessment and proactive decision making. Merely, applying single network parameter such as packet delivery [21], congestion etc. as standalone parameter can't guarantee robustness over real-time dynamic SDNs [17]. Though, a few efforts intended to apply multi-controller architecture [13]to improve fault-resilience; however, at the cost of increased latency and overheads that confine their suitability towards resource constrained (real-time) IoT/M2M communication systems. The cross-domain analysis indicates that the amalgamation of multiple dynamic network parameters for device risk assessment and proactive failure recovery strategy can be more suitable towards fault-tolerance in SDN. Yet, such multi-constraints condition might be NP-Hard problem and hence requires dynamic programming to improve failure recovery strategy while maintaining low latency. In Sync with multi-constraints decision adaptive (proactive) failure recovery method, it is inevitable to have complete network (synchronized) view [21]. This approach can not only help in applying multiple network parameters (pertaining to the deployed switches) to assess threat level and make proactive failure decision by enabling self-configuring multi-recovery path selection. However, a key challenge remains intact with classical failure recovery approaches and that is "common switch selection (CSS)" in multi-paths that might frequently cause iterative link-failure, especially under dynamic and malicious attacked networks. In other words, towards failure recovery concepts different methods [17], authors mainly use shortest path setup for recovery decision ignoring the fact that despite being shortest distance the presence of fault-prone common node might again give rise to link-failure and hence can impact overall performance. It indicates that a fault-tolerant SDN can be accomplished by deploying proactive data-layer mechanism having ability to perform risk assessment and adaptive or proactive failure recovery strategy with multi-path model having no shared components. Here, alleviating shared components or devices can help avoiding aforesaid iterative link-outage and can help disjoint failure recovery path to continue transmission without any link-outage threat. Noticeably, being proactive in nature it can apply multiple failure recovery paths, it can have multiple disjoint paths which can be selected dynamically to retain consistency and reliability of transmission without undergoing latency or computational overheads. This as a result can help achieving QoS provision to the SDN networks.

Considering above stated research gaps, challenges and allied scopes, in this paper a highly robust Heuristic Driven Self-Configuring Proactive Controller is designed for QoS-centric SDN network (HSPC-SDN). The robustness of HSPC-SDN controller can be characterized in terms of its efficacy including multiple dynamic (device) parameter driven risk assessment and heuristic driven (self-configuring) disjoint recovery path selection. More specifically, unlike classical approaches applying congestion and delay information for recovery path formation, HSPC-SDN controller applies multiple network parameters including the likelihood of successful transmission, MAC information (congestion degree, network availability) to assess efficacy of the devices to become forwarding node. Subsequently, identifying the suitable set of forwarding devices, the proposed HSPC-SDN model executes genetic algorithm (GA) which exploits source and destination information along with corresponding best forwarding nodes to decide multiple disjoint paths. Noticeably, unlike classical reactive recovery-path selection approaches which are often criticized due to high latency, our proposed HSPC-SDN model applies proactive protocol where at the initial network discovery itself it identifies the set of optimal three disjoint paths that once identifying any device failure or link-failure switches to the alternate (predefined) recovery path using AND logic function automatically. This approach helps reducing latency as well as computational exhaustion caused because of iterative network discovery and path selection. It can help achieving QoS goals. A snippet of the overall research contribution is given as follows:

- *This work contributes a highly robust multi-constraints sensitive risk assessment model that identifies the set of optimal forwarding paths to meet consistency and reliability goals in SDNs.*
- *Unlike classical single or standalone parameter driven (say, congestion, throughput, delay) failure recovery concepts, the proposed HSPC-SDN controller applies congestion information, probability of successful transmission, and dynamic link quality information to perform data-layer control decisions or forwarding decision. In addition, it applies aforesaid information in sync with source and destination information in SDN to decide multiple forwarding paths (say, failure recovery paths) that enable swift self-configuration of the network once identifying any*

*link-outage.*

- *The proposed HSPC-SDN model applies disjoint path selection based multiple failure-recovery path selection that avoids any probability of iterative link-outage and therefore helps achieving high consistency, high reliability and hence QoS/QoE assurance.*
- *Unlike classical fault-tolerant data-plane SDN controllers, the proposed HSPC-SDN model applies the concept of both device risk assessment and adaptive heuristic driven self-configuring failure recovery path selection. It makes proposed HSPC-SDN controller robust to ensure QoS delivery.*
- *The ability to exploit dynamic network parameters enables HSPC-SDN to identify vulnerable device(s) and hence would help improving future failure-proneness. In this approach only those nodes possessing superior network parameters are considered for path formation. On the other hand, the use of disjoint multiple path formation in logical-control manner helps guaranteeing consistency and hence QoS provision.*
- *The performance of HSPC-SDN model is assessed in terms of packet delivery ratio, packet loss, delay and energy consumption. The simulation results affirm that the proposed SDN controller can be vital towards major SDN-based networks including IoT/M2M, WAN, vehicular networks etc.*

The other sections of the presented manuscript are divided as follows. Section II discussed related works, followed by problem formulation in Section III. Section IV presents the proposed system and its implementation. The simulation results and allied inferences are given in Section V, while the overall conclusion is given in Section VI. References used in this study are given at the end of the manuscript.

## 2. RELATED WORK

This section discusses some of the key literature's pertaining to the fault-tolerance in SDN and different fault-resilient SDN controllers. The methods towards data-plane control mechanisms and restoration approaches are also discussed in this section. As, the use of controller towards fault tolerance imposes latency and therefore researches advocate data-plane method without applying any sophisticated controller design [16]. In this reference, authors [16] developed a mechanism that enabled switches to transmit faulty link information to the participating switches so as to alleviate traffic flooding and allied failure. Once detecting any link outage, the deployed switches broadcast a Link Failure Messages (LFM) to the relevant switches who updates their recovery paths to retain transmission; however, at the cost network discovery costs. A similar effort was made by Kempf et al. [22] who applied a monitoring function to detect link-outage in the data plane without using any controller. In this method authors communicated monitoring messages between peer switches. Noticeably, this method was designed based on delay information where in case a destination switch doesn't receive monitoring message for a defined period of 50 ms it concludes the presence of a fault in the current path. Ramos et al. [23]improved their previous contribution [24] by deploying a proactive failure recovery mechanism by exploiting the information pertaining to the alternative paths available in the packet headers. In this approach, once detecting any link outage it applies alternate paths available without indulging controller. However, in this approach authors applied alternate path estimation based on merely the VLAN and MAC Ethernet fields information and doesn't consider iterative link-failure or failure recovery under malicious attack condition. Unlike above stated approaches, authors [25] focused on achieving fault tolerance between switch and controller by applying control-traffic metrics. Zhu et al. [26] exploited backup table architecture by applying heuristic named ant-colony system to design a fault-protection system. In this approach, for individual main path, authors assigned an alternative (backup) path to assist transmission during network failure. Despite better approach; authors applied shortest path information encompassing common switch elements which may force SDN to undergo iterative link-failure in adverse condition. Gyllstrom et al. [27] developed a link-outage detection model named PCOUNT for reliable multi casting of critical Smart Grid data. Authors revealed that the use of proactive failure recovery approaches can be more effective than the reactive mechanisms. In sync with controller placement and switch migration-based multi-controller frameworks in SDNs, Al-Tam and Correia [28] and Correia and Faroq [29] suggested the concept called link-protection preplanning. Reitblatt et al. [30] on the other hand deployed fault-tolerant network programs in SDN by applying OpenFlow Fast Failover groups. In this approach, authors permitted users to assign the set of paths that the packet might employ during transmission to meet fault tolerance demands. However, the generation of the different rule-tables and group-table concept made it computationally exhaustive. To alleviate such problems, Petroulakis et al. [31] developed rule-based language to assess pattern for fault-tolerance. Cascone et al. [32]on the other hand applied finite state machines in the data plane for failure detection followed by recovery path formation. Sharma et al. [33]deployed carrier-grade networks where the route recovery was supposed to be performed within 50 ms of time. To achieve it authors applied OpenFlow Fast Fail-over groups. Similarly, authors [34] performed failure recovery for the in-band Open-Flow networks where both control as well as data traffic were broadcasted on the same channel under common operating conditions. Borokhovich et al. [35] applied classical graph model for fast forward recovery; yet failed in addressing link-vulnerability assessment and proactive fault avoidance. Van Adrichem et al. [36] applied Bidirectional Forwarding Detection (BFD) protocol [37] over each link [34] to perform fault-detection. Pfeiffenberger et al. [38] emphasized on fault-tolerant multicasting in SDN. Realizing computational overheads in recovery-path estimation, Thorat et al.

[39]applied VLAN tags as alternative path rules; yet failed in addressing network dynamism.

Network restoration has been identified as a vital technology towards fault-tolerant SDN. To achieve it, numerous efforts are made by applying the concepts of OpenFlow protocol like Fast Fail-over (FF) groups that enable resolving failure-recovery issues in data plane without deploying any sophisticated controller. In this reference, numerous efforts including the one in [40]–[44],applied a controller especially designed to assess network failure so as to identify new routes to retain transmission. However, these approaches could neither address iterative link-outage problem (especially under malicious attack conditions or multiple hardware failure) nor risk-aware fault-resilience which seem more effective towards dynamic SDNs. Kim et al. [41] applied VLANs for routing path estimation; yet, failed in addressing above indicated problems. Sharma et al. [40] on the other hand had applied FF systems by using Learning Switch, Learning PySwitch, and Routing Mode of the NOX controller [45]. Despite SDN-based fault tolerant method, the efforts of the Nguyen et al. [43] underwent prolonged route convergence time in WAN. Li et al. [44] developed a failure restoration model by exploiting a local optimal fail-over concept that mainly focused on reducing the path estimation time. Lee et al. [46] emphasized on exploiting the different fault tolerance constraints to perform adaptive path restoration. Similarly, Tajiki et al. [47] designed service sensitive failure recovery concept. Authors found that the development of a fault-aware routing can make SDN more fault-tolerant and reliable. Yuan et al. [48] designed Byzantine based fault-tolerant switches for reliable SDN. Song et al. [49] on the other hand emphasized on development of a control-path reliability concept for out-of-band controllers to deal with the data plane failures. Bhatia et al. [50] [51] on the other hand applied network coding concept to enable reliable data dissemination over SDNs. Yet, it failed in addressing iterative link-outage due to malicious attacks, switch's physical damages etc. A few recent studies like [52] and [42] indicated that an SDN might undergo congestion at certain time and hence concluding failure for it might impact overall performance. To address this problem, authors applied queue management concept with Open Flow-switch that informs the controller about congestion threshold breach so as to make proactive recovery decision. On the other hand, Kim et al. [42]exploited dynamic network traffic changes with reinforcement learning or Q-Learning method for congestion avoidance in SDNs. A similar effort was made in [53], where author's classified SDN traffic in two classes called delay tolerant and delay sensitive to perform congestion control in Mobile Edge Computing (MEC). In this approach, they stored the delay tolerant flows in MEC servers so as to prevent congestion. Recently, Bhatia et al. [50] developed a traffic congestion assessment model for SDN-based real-time urban traffic over VANET. Being a standalone network parameter based SDN, it might undergo adversaries over dynamic network condition which is quite possible in VANETs.

Apart from the above discussed approaches, authors [54] have made efforts to use multiple controllers for network monitoring and recovery assignment. Authors in [54]performed routing requirement monitoring and heuristic driven load distribution for each deployed controller. Authors in [55] developed FT-SDN by applying multiple open-source heterogeneous controllers. In case of primary controller link failure of a deployed switch it exchanges controller to make load balancing decision. Malik et al. [56] developed a fast-failure recovery concept. To achieve it, authors performed network graph partitioning on the basis of the node's similarity. In case of any failure, the failure recovery model considers a specific community to reroute the data rather opting alternative path computation so as to reduce computational cost. A proactive fault-handling model was proposed for SDN in [17] in which SDN controller monitors network-flow disruption to make forwarding decisions. Authors in [57] focused on estimating the most reliable path for SDN data forwarding. Interestingly, this approach stated that the disjoint paths can't be optimal and hence suggested to have reliable path as backup, where they defined reliable path as the one with maximum number of shared components in between source-destination pair. Noticeably, authors forgot to consider that employing more shared component in backup paths can even undergo iterative network failure due to common share component or switch in SDN. Moreover, authors [57] considered merely one backup path for routing that might become inferior over any iterative network failure over common shared component. Recently, authors in [58] developed Repair Path Refinement with Destination based Tunneling (RPR-DT)) for SDN Candidate Selection (SCS) by applying shortest repair reactive path. Here, they focused on reducing the number of SDN switches between source-destination pair. Authors in [59] developed a failure recovery model named SafeGuard [59] for SD-WAN by applying bandwidth and switch memory utilization. It applied FF group, in addition to the alternative link capability before rerouting the failed transmission. Authors in [60]developed a Multi path Resilient Routing Scheme for SDNs-enabled Smart Cities Networks (MPResiSDN). Despite efforts, authors failed in assessing or employing the best forwarding path and the number of rerouting paths. Lin et al. [61] developed a switchover concept to address link failure problem in SDN. Here, they designed switchover as a programmable function that swaps the action buckets and reconfigured the less congested path to retain transmission. Efforts like CORONET and Self-Healing Protocol (SHP) focused on automatic switch mechanism to make SDN delay resilient while addressing failure recovery. Heuristic methods like Breadth First Search (BFS) were applied in [62]to perform failure recovery in SDN control. A recent technology exploiting the Shared Risk Link Groups (SRLGs) was applied to employ the relationships of complex failures to achieve failures avoidance in SDN environment. However, the use of classical Column Generation and Bender Decomposition make it computationally exhaustive to avoid SRLG failure. Kiadehi et al. [63]on the other hand applied SRLG for non

overlapping path estimation for backup path identification. Authors applied Dijkstra and Disjoint Path (DP) algorithm for back-up path estimation and ignored common point failure problem in multi-path formation. Authors in [64] applied SRLG for two disjoint paths between the source and destination switches to retain transmission.

## 3. FAULT TOLERANCE: THE KEY TO PERFORMANCE CONSISTENCY AND RELIABILITY IN SDN-BASED SYSTEMS

As discussed in the previous sections, the foundation of performance consistency and reliability roots in how fault-tolerant forwarding decision is made. On the other hand, unlike multiple controllers driven SDNs, single SDN controller can be more suitable to meet performance demands. In sync with these two facts, developing either a single SDN controller driven approach or controller less forwarding or fault tolerant proactive routing can be vital. To be noted, in major SDN-based system, delay turns out to be the decisive factor impacting overall performance. This is because, once undergoing any link-outage or failure the classical SDN controllers have to undergo fault detection, recovery path formation and recovery path reassignment. These overall processes introduce significantly large delay and hence impacts overall QoS performance. To alleviate such issues, developing proactive failure recovery can be of vital significance. In sync with this motive, this work intends to develop a robust proactive fault-tolerant data-plane controller model for SDNs. Before discussing the proposed controller architecture, discussing the key issues of performance consistency and reliability aspects in SDN is vital, as it can help addressing at hand problems while designing the protocol. A snippet of the key performance consistency issues and allied challenges is given as follows.

### A. *Single SDN Controller Driven Model and allied Performance*

Typically, in a single controller driven SDN, once any deployed switch or node initiates transmission or broadcast request and generates a new flow, the controller is expected to perform the following tasks.

- *To initiate transmission, a traffic flow or packet signifying a new flow is obtained at the network ingress switch, which is subsequently forwarded to the corresponding controller.*
- *Once receiving the new-flow request from the switch, controller then assess the request and estimates the forwarding path for that specific flow-request by following network policies and network dynamic information.*
- *Estimating the forwarding path, the controller updates to the requesting switch about its future forwarding path and updates path to the flow-table also called network information base (NIB) of that switch.*
- *Receiving the forwarding path, the switch initiates transmission sand continues it till the complete data traffic is not transmitted to the destination.*

In sync with above discussed functional paradigm a few key components play decisive role. These are: What are the Network Policies used to estimate forwarding path (s)? What specific network parameter(s) the controller consider to identify forwarding path? Does the controller verify suitability of the intermediate switches or nodes to become forwarding node or switch? If not, would not the inferior or malicious switch (say, fault-prone switch) can make entire transmission fail? Does the controller address the problem of iterative link failure or outage, which is common in case of certain malicious attack cases or hardware failure? What is the failure recovery policy of the controller to ensure QoS delivery in SDB-based solutions? In addition to these key questions, a most important aspect of SDN implementation is that whether the deployed controller is capable of ensure failure-recovery within 50ms of delay to meet QoS/QoE demands [6] [9]? This is because if the deployed controller is not capable of handing the failure recovery within a small span, it might impact overall QoS/QoE performance. This research intends to achieve a robust fault-tolerant data-plane controller solution which could fulfill the overall QoS/QoE expectations while guaranteeing optimal answer for the above stated questions. Noticeably, in this research to alleviate any possible delay the focus is made on designing a fault-tolerant proactive failure recovery approach with multi-path self-configuring capability. Moreover, to guarantee that the switches or nodes participating the forwarding paths don't lead any future failure, the proposed work performs multiple parameters driven risk assessment model. Subsequently, it applies heuristic-based forwarding path estimation model for SDN systems.

As stated in the previous sections, this research addresses multiple key challenges of SDN-based system including risk assessment and proactive routing decision for fault-tolerant failure recovery model for SDN. More specifically, this research intends to address delay-resilient failure recovery so as to ensure QoS/QoE performance. In this reference, the proposed Heuristic Driven Self-Configuring Proactive Controller for QoS-centric SDN network (HSPC-SDN) at first performs device/node profiling followed by failure-recovery to meet QoS demands. In almost all existing failure recovery approaches, authors ignored the fact that inclusion of a fault-prone node or switches can cause network failure during transmission, which can also cause iterative link-outage and hence can impact QoS severely. To alleviate such problem, the proposed HSPC-SDN model at first focuses on identifying a set of best forwarding nodes or switches better source-destination pair to fulfill QoS demands. To achieve it, it performs risk assessment and device profiling concept. Once receiving broadcast or traffic flow request the data-layer model executes network policy to identify the reliable forwarding nodes or switches. In sync with data-plane architecture, the proposed model collects different MAC layer information including conges-

tion information, IEEE 802.15.4 MAC values, probability of successful transmission, link quality information etc. to perform network risk assessment, also called node profiling. To achieve it, the controller transmits multi cast to the relevant nodes between source-destination pairs and exploits above stated key parameters. In this manner, the aforesaid parameters (i.e., congestion information, IEEE 802.15.4 MAC values, probability of successful transmission, link quality information) represent the key decision values for forwarding path estimation. Once selecting the best forwarding nodes, the proposed HSPC-SDN model executes a heuristic model named GA to estimate multiple disjoint paths in between source-destination pair. Noticeably, here it focused on estimating the disjoint paths with no shared elements so as to alleviate any probable link-failure due to common point failure [65]. Since, it is a NP-hard problem, it applied GA algorithm that estimated three different disjoint paths in between source and destination. Thus, once identifying any link-failure, the proposed model executes logical-AND function to select alternate recovery path Noticeably, unlike reactive failure recovery concepts in which authors mainly applied reactive concept where it starts recovery path estimation after detecting node or switch failure, the proposed HSPC-SDN model applies proactive concept in which it decides multiple recovery paths at the time of network discovery for each source-destination pair. The multiple paths identified are stored in NIB which are selected automatically as the failure-recovery path (using logical AND function) once detecting any link-outage. In this manner, the proposed HSPC-SDN model provides a self-configuring failure recovery concept without imposing any latency which is vital towards QoS-centric communication. Moreover, the inclusion of risk assessment and node profiling enabled forwarding paths to be fault-tolerant to meet QoS demands, especially the consistency and reliability. Here, the proposed model applied multiple network parameters to assess forwarding nodes and hence it avoided any possibility of false positive that helped retaining optimal routing decision for SDN-based systems. Being proactive in nature, the proposed model can be applied as a single controller based SDN or can directly be implemented as data-plane middle ware to reduce delay in SDN-based systems. The detailed discussion of the overall proposed HSPC-SDN model and its implementation is given in the subsequent section.

## 4. SYSTEM MODEL

As stated in the previous sections, this work focused on both risk assessment and potential forwarding device or node identification, as well as heuristic driven disjoint multi-path estimation to meet QoS-centric SDN demands. Thus, the overall research model encompasses the following two key phases:

1) Multi-constraints Node Profiling and Network Information Base (NIB) Formation, and
2) Heuristic-Based Disjoint Multi-Paths Estimation with No-Shared Component for Failure Recovery.

The detailed discussion of these key methodologies is given in the subsequent sections.

### *4.1 Multi-constraints Node Profiling NIB Formation*

In real-time SDN-based environment such as VASNET, WAN, IoT/M2M communication etc., the participating switches or nodes might undergo dynamic network characteristics including change in congestion, link-quality and even packet delivery rate. These network behaviors or parameters might undergo severe dynamism under exceedingly high changing topology, and hence merely applying reactive data-plane control strategies can't yield expected performance. On the contrary, such networks might undergo frequent network-failure and hence scheduling based on reactive approach might cause significant latency that might adversely impact the overall QoS/QoE performance. Majority of the existing SDN control methods or allied network policies where authors have directly applied reactive or proactive routing concepts once identifying faults without assessing suitability of a participating node to become a member of forwarding node or switch. Noticeably, randomly selecting a neighboring node or device (say, switch) as forwarding node based on either distance formulation or merely because it exists in between source-destination pair can make entire network more vulnerable. Considering this fact, in this work before performing failure recovery policy or forwarding path estimation, the proposed HSPC-SDN model performs risk assessment of each participating node by exploiting their (node's) corresponding dynamic network behavior. Once receiving transmission request from a switch, HSPC-SDN model transmits the multi cast and obtains the node's key parameters including IEEE 802.15.4 MAC information, the probability of successful transmission, congestion information and link-quality information. To be noted, the proposed HSPC-SDN model applies multiple network parameters to ensure reliability of the nodes or switches to become the potential forwarding node. Once identifying these network parameters, HSPC-SDN model shortlist the best suitable set of nodes or switches in shortest distance manner to perform forwarding path formation or disjoint multi-path estimation, which is discussed in the subsequent sections. A snippet of the key network parameters applied towards potential forwarding node estimation is given as follows:

### *4.2 MAC Information*

Some of the existing SDN Controllers have merely applied delay and congestion information to execute failure-recovery task. These approaches hypothesize that the non-linear MAC information indicates node failure. However, there are numerous cases including contention, hardware malfunction and malicious attacks, where despite active a node can be labeled as faulty node of failed node. This as a result can cause redundant transmission giving rise to the QoS compromise. To alleviate such issues, applying multiple node parameters can enable more accurate deci-

sion making. In sync with delay resilient failure recovery demand, the proposed HSPC-SDN model measures MAC information of each node. To achieve it, the controller multi casts HELLO beacon message and receives acknowledgment from each participating node. Once obtaining the ACK message as uni cast, HSPC-SDN model measures the different node parameters including the probability of successful transmission, link-quality and congestion.

*1) **Probability of Successful Transmission***

In the proposed model, the controller measures the probability of successful transmission by the participating node Z . The probability of successful transmission is estimated as per the equation (1).

$$P_M = \frac{\xi_{RX(t_{i-1},t_i)}}{\xi_{Exp(t_{i-1},t_i)}} \tag{1}$$

*2) **Dynamic Link Quality***

In addition to the successful transmission probability, HSPC-SDN estimates the link quality in between node pairs by using equation (2). To achieve it, it applies the statistical information including the total number of packets transmitted and the number of packets received during ($t_{i-1}$, $t_i$) period. In sync with dynamic link quality assessment, we applied Moving Window Link Estimation approach, defined in (2).

$$\beta_{DLQI} = \mu * \beta_{DLQI} + (1 - \alpha) * (PDR_{ij}) \tag{2}$$

In (2), (PDRij) being the packet delivery ratio in between the two nodes i and j is estimated as per (3).

$$PDR_{ij} = \frac{P_{Rx}}{P_{Tx}} \tag{3}$$

In (3), $P_{Rx}$ be the total number of packets retrieved, while $P_{Tx}$ be the total number of packets transmitted by $i$-th switch to the $j$-th switch or sink. In above derived equation (2), the parameter $\beta_{DLQI}$ states the dynamic link quality between $i - j$ node pairs, while $\mu$ states the network coefficient varying in the range of 0 to 1. Noticeably, the value of $\mu$ depends on the network condition where a large value signifies better network environment with low loss probability, while the lower value of $\mu$ indicates high disturbance in the network.

*3) **Cumulative Congestion Degree***

Undeniably, a large number of SDN-based systems undergo continuous data transmission with non-linear traffic patterns and mobility. For instance, SDN-driven M2M and IoT ecosystems often undergo non-linear traffic patterns, where the severity of congestion increases over dynamic topologies and non-linear transmission behavior. Specifically, in dynamic topology driven ecosystems where the switch(es) can be in mobile state as well can undergo sudden increase in payload and hence congestion. In addition, unlike wired network structure, wireless networks, especially driven by wireless sensor networks (WSNs) or low power lossy networks (LLNs) might undergo congestion condition. Being greedy in nature, the likelihood of getting congested becomes a common problem in WSN/LLN driven SDN systems. Practically, the congestion on a node i can be caused because of multiple neighboring nodes, trying to transmit their data through i-th node. Because of this reason, this paper defines congestion as the cumulative congestion, as a dynamic network parameter signifying the extent to which the participating switch or node is congested (i.e., the extent to which the resource is being employed). Considering dynamic network characteristics, the proposed HSPC-SDN model introduced a parameter called cumulative congestion degree (CCD). To be noted, in sync with real-time congestion resilient transmission over SDN, we hypothesized each switch to have two distinct kinds of buffers called real-time buffers (RTB) and non-real-time buffers (NRTB). Here, RTB is allotted to store real-time traffic including mission critical data logs, instruction sets etc., while NRTB is dedicated to store non-real time traffic signifying multimedia data or the data to be merely stored for future analysis. Here, the key motive was to provide fair resource provision and allied scheduling to meet QoS/QoE- demands. In sync with above device configuration or resource capacity, we exploited two key information; the maximum buffer capacity and the current available buffer capacity of RTB and NRTB buffers, concurrently to estimate CCD. In this paper, equation (4) and (5) were applied to estimate CCD for each candidate forwarding node.

$$CCD_i = \frac{CD_{RTB} + CD_{NRTB}}{CD_{RTB-Max} + CD_{NRTB-Max}} \tag{4}$$

$$CCD_{FN} = \sum_{i=1}^{N} CCD_i \tag{5}$$

In equation (4), CRRTB and CDNRTB state the buffer available over RTB buffer and NRTB buffer, respectively. The other variables CDRTB,Max and CDNRTB,Max signify the maximum usable buffer with RTB and NRTB, correspondingly. The proposed HSPC-SDN model estimates CCD of each candidate node for the period of (ti-1,ti) so assess future congestion likelihood and alleviate any possible future congestion in the path. Here, the key motive is to ensure data delivery without causing any congestion probability.

*4) **Traffic Non-linearity***

SDN-based systems might undergo non-linear traffic condition due to change in payloads, congestion, topological changes etc., especially in mobile-switch driven IoT SDN systems. Moreover, due to high network dynamism the node might undergo flooding or packet drop. Especially in case

of network attack condition the malicious node can cause burst transmission causing flooding. Under such dynamic conditions, labeling a node or switch as failed or faulty can force the network to undergo disturbed QoS- performance. In sync with this fact, the proposed HSPC-SDN model considered traffic overflow or non-linearity condition as behavioral parameter to assess suitability of a switch or node to become forwarding node. Here, we hypothesized that a node with exceedingly high overflow or non-linearity might cause packet drop and hence reduced performance. To achieve it, HSPC-SDN model estimated a parameter called queue length at the MAC. This information is multi cast as ACK to the neighboring nodes including controller, so as to make adaptive decisions. Let, i be the participating switch in source-destination path and lj states j-th be the queue length during the assessment period (ti-1,ti) . Now, over the queue-length of L, it measured the average traffic load at a candidate switch using equation (6).

$$T_{load_i} = \frac{1}{L}\sum_{j=1}^{N} L_j \tag{6}$$

Consider that $l_max$ be the maximum possible queue length at a switch buffer, it estimated the cumulative traffic density (CTD) as per the equation (7).

$$T_{loadDens_i} = \frac{T_{load_i}}{l_{max}} \tag{7}$$

In this manner, the likelihood of successful transmission for a participating node i, $P_succ_i$ is estimated as per the equation (8).

$$P_{succ_i} = [1 - T_{loadDens_i}] \tag{8}$$

Recalling the fact that the likelihood of successful transmission is directly related to the packet delivery, and therefore with low successful transmission the probability of re transmission can increase causing high latency and resource exhaustion. Considering this fact, the proposed HSPC-SDN model considered only those switches with minimum queue length to become the forwarding node so as to ensure fault-resilient transmission in SDN-based systems. In addition to the above discussed behavioral pattern, the proposed model derived a parameter called link-quality change index (LQCI). A snippet of this parameter estimation model is given as follows:

*5) **Link Quality Change Index***

In dynamic or non-linear network conditions, overhearing can be a key problem forcing participating nodes to undergo redundant signaling or allied transmission costs. To ensure cost-efficient and delay-resilient transmission (for QoS) over SDN-based systems, the proposed model derived a parameter called LQCI, signifying link-trustworthiness of a participating switch or node. Mathematically, we applied equation (9) to estimate LQCI for a node i, as:

$$\eta_i = \gamma_i + \delta_i \tag{9}$$

In (9), $\gamma_i$ represents the ACK arrival rate, while the link-outage frequency at the node i is given by $\delta_i$ . Here, we hypothesize that for a node or switch to become reliable or trustworthy, the maximum rate of arrival $\gamma_{iMax}$ is required to be same as the rate of link-outage. Therefore, the highest link outage ($\delta_{iMax}$)is estimated as per (10) [36].

$$\gamma_{i-Max} + (\delta_{i-Max}) = 2.\sigma_i \tag{10}$$

Thus, applying above statistics, we derived LQCI as per (11).

$$\eta = \frac{(\gamma_i + \delta_i)}{(2.\sigma_i)} \tag{11}$$

In this manner, retrieving the value of (10), the proposed HSPC-SDN re-estimated the probability of successful transmission as per the equation derived in (12).

$$P_\eta = 1 - \eta \tag{12}$$

To ensure reliable transmission over the deployed SDN-based systems, our proposed HSPC-SDN model considers only those devices having low LOCI to perform forwarding path selection.

Once estimating the node profile values as defined in equations (2), (4), (8) and (12), our proposed HSPC-SDN identifies as set of most suitable device to perform forwarding path selection. To achieve it, it follows the criteria derived in (13).

$$\begin{aligned} Trust - Node_{sel} = f[&(max\beta_{DLQI}),\\ &(minCD_r), (maxP_{succ_i}), (minP_\eta)] \end{aligned} \tag{13}$$

This is the matter of fact that the use of (13) can yield reliable and fault-tolerant transmission; however, doesn't address the failure-recovery aspects. Considering this motive, in this paper we focused on identifying a set of multiple paths between source-destination pair to ensure delay resilient and fault-tolerant failure recovery concept for QoS communication in SDN-based systems. Though, in majority of the existing approaches, authors have applied single network parameters as discussed above (13) to perform recovery path estimation, in addition to the delay-based methods of distance-based approaches. However, none of the existing method addressed the likelihood of consecutive node or allied link-failure due to common node failure

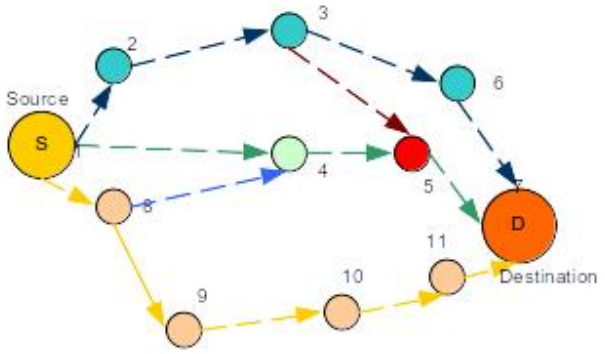


Figure 2. Fault tolerant multi-path transmission setup

Consider that the deployed SDN-based network be Fig.2. Let the nodes N =N1, N2,N3,N4,N5,N6,N7,N8,N9,N10,N11 be the selected forwarding nodes, where the nodes N1 and N7 be the source node and the destination nodes, respectively. In this manner, let there be the multiple possible (say, candidate paths) paths be the following:

$$\begin{aligned} Path1 &= N1 \rightarrow N4 \rightarrow N5 \rightarrow N7 \\ Path2 &= N1 \rightarrow N4 \rightarrow N5 \rightarrow N7 \\ Path3 &= N1 \rightarrow N4 \rightarrow N5 \rightarrow N7 \\ Path4 &= N1 \rightarrow N4 \rightarrow N5 \rightarrow N7 \\ Path5 &= N1 \rightarrow N4 \rightarrow N5 \rightarrow N7 \end{aligned} \tag{14}$$

As depicted in equation (14), with the illustrated network deployment there can be a total of five different paths, where Path1, Path4 and Path3 seem to be selected based on shortest distance path formulation, as applied in numerous existing works. Therefore, those routing models or network policies (say, failure recovery policy) applying shortest path routing would select Path1, Path4 and Path3 based on lower inter-source-destination distance. However, these existing approaches don't consider the failure of node N5, which is the common node for all these paths, can collapse entire networks and hence neither of these defined paths can be successful in delivering data successfully. Let, Path1 collapses or undergoes network failure, as per policy the SDN controller is supposed to switch to the Path3 or Path4; however, post initiating the alternate path it might again undergo link-outage due to disruption at the common node N5. In other words, the multiple forwarding paths or recovery paths with common nodes (say, shared component(s)) might undergo iterative network failure impacting overall QoS performance [66]. Therefore, for a fault-tolerant and reliable data transmission it is must to select multiple forwarding path or recovery paths while ensuring no common node or shared component [66].

Considering above stated issues and allied scopes, in this paper we designed HSPC-SDN model in such manner that once estimating the set of best forwarding nodes or candidate switches, it executes network policy for disjoint multiple forwarding path estimation. Noticeably, being a proactive routing concept, our proposed model applied NIB information where the dynamic node characteristics or node information are stored and updated dynamically. Thus, our proposed HSPC-SDN model exploits dynamic information of the potential candidate nodes to perform disjoint forwarding path estimation. To be noted, the at hand disjoint recovery path estimation or multiple (disjoint) path estimation is a NP-hard problem, and hence require heuristic to solve it. To achieve it, in this paper we applied GA, a well-known evolutionary computing and heuristic approach to estimate the disjoint forwarding paths. The detailed discussion of the proposed Heuristic driven disjoint recovery path estimation is given in the subsequent section.

***A. Heuristic-Based Disjoint Multi-Paths Estimation with No-Shared Component for Failure Recovery*** In sync with high consistency, availability and reliability (say, fault-tolerance), the proposed HSPC-SDN model executes GA algorithm over the dynamic information pertaining to the selected nodes in NIB. Here, the key objective of GA is to identify the set of three disjoint paths with no shared components or common switches. However, unlike classical methods were merely delay and congestion information were applied as the criteria for recovery path estimation, we obtained link-connectivity, node or device availability and distance information to perform disjoint path formation. To achieve it, our proposed HSPC-SDN model at first performed first order approximation so as to identify the nodes or path unavailability. Here, we defined path unavailability as the addition of the unavailability of all comprising nodes or switches in targeted source-destination paths. Thus, HSPC-SDN model executed Monte Carlo simulation which helped in estimating the dynamic topology and corresponding network information update in NIB. Moreover, it enabled probabilistic network deployment. To be noted, in this work we deployed complete network as per Bayesian network model over the defined SDN-based environment. This as a result helped estimating the network parameters (i.e., link connectivity, link-unavailability etc.) proactively to make dynamic routing decisions.

***B. Link Connectivity Estimation*** In the considered SDB-based system, we define node connectivity as the probability that at least one forwarding path would be active or present in between the targeted source-destination pair. In this reference, a switch S0 remains connected to the failure recovery path when S0 is active, provided that at least one path exists to connect source-destination pair. In HSPC-SDN model, we assume that each participating switch or device possesses two disjoint forwarding paths, with no common or shared component(s). Now, let the forwarding paths for switch $S_0$ be the $\mathbf{P_0}$, $\mathbf{.....P}_k - 1$, while $\bar{P}_k$ be the possible connectivity with Pk. HPSC-SDN estimates the connected path using (15).

$$C(S_0) = A(S_0)A\left(\bigcup_{k=0}^{k-1}\bar{P}_k\right)A(C) \tag{15}$$

In above derived function (15), states the set of possible paths available to meet failure recovery task. In sync with the real-time network dynamism, despite being active the connectivity of $S_O$ may undergo network failure, and hence link-loss in case the path $\bar{P}_k$ fails due to certain quantifiable reasons (i.e., node death,malicious attack, physical damage etc.). Therefore, assuming that the deployed switch and corresponding link condition are independent, HSPC-SDN calculates corresponding link-availability using (16).

$$A\left(\bigcup_{k=0}^{k-1}\bar{P}_k\right) = 1 - \Pi_{k=0}^{k-1}U_r\left(\bar{P}_k\right) \tag{16}$$

Let, the deployed SDN-based models be encompassing $S_{k_0}$,.....$S_k$,$f_k$ devices or switches, with their respective link in path k as $e_{k,1,2.....}$,$n_k$,$f_{k-1}$,$f_k$ , respectively. In this case, the link-availability is calculated as per (17).

$$\begin{aligned} U_r\left(\bar{P}_k\right) &= 1 - A_r\left(\bar{P}_k\right) \\ &= 1 - \Pi_{i=1}^{f_{k-1}}A_n\left(S_{k,i}\right)\Pi_{j=0}^{f_{k-1}}A_e\left(e_{k,j,j+1}\right) \end{aligned} \tag{17}$$

Thus, applying above derived network availability scenarios (14-17), HSPC-SDN model estimates the link connectivity for a transmitting switch SO using (18).

$$\begin{aligned} C(S_0) = A(S_0)A(C) * (1 - \Pi_{k=0}^{k-1}(1 - \Pi_{i=1}^{f_{k-1}}A_n(n_{k,i}) \\ \Pi_{j=0}^{f_{k-1}}A_e(e_{k,j,j+1}))) \end{aligned} \tag{18}$$

In sync with the fault-tolerance motive, our proposed HSPC-SDN model executed GA in such manner that it considers forwarding (multiple) paths with high connectivity and no-shared component. The above derived functions (14-17) helped estimating linkconnectivity information of each participating candidate node.

To estimate the disjoint paths with no shared switches or devices, HSPC-SDN assume that the disjoint connectivity can be accomplished by decoupling the significance of shared devices from the associated links or paths. Now, let R0 and R1 be the two forwarding paths, then the corresponding link-connectivity can be derived as per (19).

$$\begin{aligned} C(S_0) = \; & \Pi_{j\in\phi_s}^{f}A_n(n_j)\Pi_{k\in\phi_e}^{f-1}A_e(e_{k,k+1}) \\ & \times(1 - (1 - \Pi_{i\in\phi_{s,0}}A_s(S_{0,i})\Pi_{j\in\phi_{e,0}}A_e(e_{0,j,j+1} \\ & \times(1 - \Pi_{i\in\phi_{s,1}}A_s(S_{1,i}) \\ & \Pi_{j\in\phi_{e,1}}A_e(e_{1,j,j+1}))) \end{aligned} \tag{19}$$

In above derived link-connectivity function (19),$\phi_s$ and $\phi_e$ represent the shared devices. Similarly, the set of disjoint devices over i-th path are given by $\phi_s, i$ and $\phi_e, i$.Thus, applying aforesaid first order approximation,in the form of unavailability, it estimates the connectivity loss as per (20).

$$L(S) = 1 - C(S_0) \tag{20}$$

$$\begin{aligned} L(S_0) \cong \; & \textstyle\sum_{j\in s}^{f}U_s(S_j) + \sum_{k\in\phi_e}^{f-1}U_e(e_{k,k+1}) \\ & +(\textstyle\sum_{i\in\psi_{e,0}}U_s(S_{0,i}) + \sum_{j\in\phi_{e,0}} \\ & U_e(e_{0,j,j+1})) \times (\textstyle\sum_{i\in\phi_{s,1}}U_s(S_{1,i}) + \\ & \textstyle\sum_{j\in\phi_{e,1}}U_e(e_{1,j,j+1})) \end{aligned} \tag{21}$$

reference to (21), we estimated the likelihood that the disjoint path doesn't impact the likelihood of retransmission (22).

$$L(S_0) \cong \sum_{j\in\phi_s}^{f}U_s\left(S_j\right) + \sum_{k\in\phi_e}^{f-1}U_e\left(e_{k,k+1}\right) \tag{22}$$

Considering above derived equations (21-22), it can be inferred that the forwarding path pair can be designed without applying any shared component or device.Therefore, obtaining the link connectivity (say,availability) as the cost function in GA, our proposed HSPC-SDN performed disjoint path estimation.

***C. GA- Driven Disjoint Path Estimation*** GA algorithm is an adaptive search method based on the evolutionary concepts of natural selection that tends to identify the optimal or sub-optimal solutions from multiple available subsolutions or the set of solutions. Functionally, GA algorithm possesses three key steps,population initialization, crossover and mutation. Here, population signifies the set of solutions (i.e., set of paths connecting source and destination nodes or switches).These solutions represent the chromosome possessing a form of binary strings in which all allied features or factors are encoded. Once initiating the random population generation, GA algorithm estimates the fitness value, also called fitness function for each chromosome.The fitness value represents a user-defined function that provides the estimation results for each chromosome, and therefore a higher fitness value represents the chromosome to be the leading one. In our proposed work, we considered link-connectivity, number of hops with no shared components as the objective function or the fitness function (22). i.e., the proposed model exploits link connectivity or network availability, number of hops (between source and destination) with no-shared component for each possible path to assess its fitness to become the forwarding path solution. Thus, GA is executed over each candidate path and solution optimization continues iteratively by adding a new hop device. This mechanism continues until the probability of getting a superior path becomes very low. In our applied GA

model, we employed two iterative processes, path selection and pruning. Once identifying the forwarding path over an iteration k the other path(s) from $S_k$ having low-cost function or poor link connectivity are pruned. In this work, we applied (23) as the objective function **c(P)**to estimate fitness of each candidate path **P**.

$$P^* = arg\min_{R} c(P) \tag{23}$$

Consider that $\bar{P}$ be the forwarding path with devices possessing zero connectivity loss. Then, the path R can be connected to the node $S_f$, and therefore for any forwarding path $M_i \in S_k$ , L($\bar{P}$,$M_i$)be the connectivity-loss in reference to the source switch $S_O$. In this manner, the average connectivity loss is obtained as per (24)

$$\tilde{L}(P) = \frac{1}{S_c}\sum_{i=1}^{S_c} L\left(\bar{P}, M_i\right) \tag{24}$$

In our proposed work, we redefined the cost function (22) as per (25), where E(P) was estimated on the basis of the mean loss imposed per link across the paths (i.e., dual disjoint paths). We estimated E(P) using the mathematical model as defined in (26).

$$c(P) = \tilde{L}(P) + E(P) \tag{25}$$

In other words, E(P) is estimated as per (25).

$$E(P) = \frac{1}{N_c}\sum_{i=1}^{N_c} E(P, M_i) \tag{26}$$

where

$$E(\bar{P}, M_i) = \frac{\tilde{L}(M_i)}{\lambda} d(S_p, S_f) \tag{27}$$

As already stated, heuristic driven disjoint forwarding paths selection model applied bot link-connectivity as well as low hop counts (with no shared components) as objective function. Therefore, to calculate the inter-node distance values, we applied graph theory concept. Here, network graph represents a graph matrix A having the different devices $a_{ij}$ with status $a_{ij} = 1$ signifying that the link connectivity between the node i and j is active.Otherwise, it considers the conditions defined as $a_{ij}$ =0 and $a_{ij}$ =1. Thus, it obtains a matrix B(k), defined as (28).

$$B(k) = A^k \tag{28}$$

In (28), B(k) encompasses $b_{ij}$(k) which is equivalent to the total paths to reach the destination switch j from the ith switch, while maintaining the hops lower than k. In this manner, with $b_{ij}$(k)=0 there would not be the other path connecting j from i device in k-hops. Here, the distance between i to j device was obtained as per the shortest path formulation, defined in (29).

$$d(i, j) = \min_{b_{i,j}k>0} k \tag{29}$$

The above derived model (29) signifies that the distance information d(i,j) can have the minimum k hops when $bij(k) > 0$. Thus, applying the cost functions as derived in (21) and (29), our proposed HPSC-SDN model identified three disjoint forwarding paths with strictly no shared component. Noticeably, once estimating the values of (29), the identified forwarding paths available in $S_k$ are updated proactively in NIB. Noticeably, in the proposed model, the proposed heuristic driven forwarding path selection model identifies three different disjoint paths for a transmission pair ij, and updates those paths as $Path_{ij}^1$,$Path_{ij}^2$and $Path_{ij}^3$. These disjoint paths are stored in NIB that the controller use to select recovery path proactively. During run-time transmission in case a controller (here, HSPCSDN) identifies any link-outage in ongoing path, it switches to the other alternate failure recovery path(s)using AND logical function and retains disrupted transmission to meet QoS demands. In this manner, unlike classical reactive failure recovery approaches HSPC-SDN model identifies three sets of forwarding paths for a targeted ij transmission request and applies AND logic to select failure recovery path once detecting any link-outage in run-time. This self-configuring ability not only reduces network rediscovery cost but also minimizes latency to meet QoS demands. Noticeably,the proposed HSPC-SDN model was designed in such manner that it can be applied as a single controller solution or can also be employed as data-plane middleware on each device to ensure fault-tolerant QoS performance.

## 5. Results and Discussion

In this paper, a robust Heuristic Driven Self-Configuring Proactive Controller (HSPC-SDN) and/or failure-recovery model is developed for SDN-based systems. Unlike major at hand solutions, this work focused on alleviating any fault-probability due to inferior device (say, switch) characteristics and delay resilient proactive failure recovery strategy to guarantee QoS delivery. The development of this model was hypothesized on the three facts, given as:

- $H_{01}$:*Including risk-free devices for forwarding path estimation can achieve fault-tolerant transmission in SDN-based systems.*
- $H_{02}$:*The use of proactive multi-path selection and adaptive self-configuration can enable delayresilient transmission in SDN-based systems.*
- $H_{03}$:*The implementation of disjoint multi-path selection with no shared component can enable fault-*

*tolerant failure-recovery and QoS-centric transmission in SDN-based systems.*

In sync with above stated hypotheses, we designed the proposed HSPC-SDN model in a multi-phased schematic. Towards first hypothesis, HSPC-SDN model applied multiple dynamic network parameters to assess suitability of the devices or nodes to become forwarding nodes. Here, the key motive was to consider only those switches or devices (say, nodes) which possess superior node characteristics and can enable reliable communication for a target flow. In this reference, HSPC-SDN measured different MAC related data-plane information encompassing dynamic link quality,cumulative congestion degree, probability of success transmission and link-quality change index to select the suitable set of forwarding nodes for a traffic flow (i.e.,source driven traffic towards the target destination). To reduce iterative computational costs, especially during fault-discovery or detection, recovery path estimation and recovery path assignment, we considered the concept of proactive failure recovery for which an NIB was taken into consideration. Here, NIB (Network Information Base) helped estimating dynamic network information and process it to select suitable set of forwarding nodes for each requesting traffic flow. Thus, implementing this approach, HSPC-SDN model ensured that no faulty or fault-prone device is considered in forwarding path. This as a result intended to alleviate any fault probability caused due to pre-existing hardware malfunction,malicious attacks etc.

Once identifying the set of optimal forwarding paths,HSPC-SDN model intended to perform proactive forwarding path selection. The motive behind multiple forwarding path selection is to reduce delay to serve QoS/QoE demands. The proposed HSPC-SDN model designed forwarding path cum failure recovery path estimation in such manner that only those paths possessing higher availability or connectivity (to meet consistency demands) with lower inter-node distance and strictly no common (or shared) devices are selected as the optimal path selection. To achieve it, our proposed model applied GA algorithm which exploited the NIB information and allied dynamic path information (including hops information, connectivity, availability) to estimate three distinct forwarding path that also serves as the failure recovery path. Recalling the fact that heuristic methods often impose computational overheads, we applied GA with merely 10 input (random) population so as to get three sub-optimal (best and disjoint) forwarding paths. Moreover, GA parameters like crossover and mutation probability were considered as 0.6 and 0.4,respectively to maintain low computational overheads.To estimate dual disjoint path selection as the recovery path or the forwarding path, the proposed GA model applied link-connectivity, number of hops with no shared component as the objective function. Noticeably,the aforesaid optimization condition represents an NPhard problem and hence the use of heuristic model GA was justifiable. Unlike other heuristic such as ant colony system [26] or Breadth Search First [65] GA is lightweight and hence consumes lower computational cost. It makes proposed system more adaptive towards real-time network decision making. Thus, the overall proposed model selected three different disjoint paths for a source-initiated transmission request. These transmission paths are stored in NIB, a proactive network management table which is maintained by the single SDN controller HPSC-SDN; though, the overall proposed routing model can be applied as a middleware on each participating node. Functionally, the proposed model uses a single shortest distance path (with the minimum number of hops) to transmit the data from source to the destination. Once identifying any link outage in currently operating forwarding path, it executes failure recovery path which are stored in NIB table. Here, HPSC-SDN model applies logical AND function to select the recovery path. This process is continued till the complete source data is transmitted to the destination node successfully. Here, the use of automatic AND-logic driven recovery path selection reduces the time of forwarding path discovery, route estimation and assignment and hence alleviates any likelihood of delay. This as a result enables consistent transmission with minimum or no retransmission probability and hence low delay and energy consumption. It makes proposed HSPC-SDN model more suitable towards resource constrained SDN networks such as SDN-based IoT/M2M or WAN systems.

Before discussing the statistical performance characterization of the proposed HSPC-SDN model, a brief of the recent and closely designed fault-tolerant SDN methods is given in the subsequent section. Here,the key motive is to identify design specific and problem specific robustness or superiority of the proposed HSPCSDN model over the other existing approaches. This discussion can be called as qualitative assessment to identify superiority of the proposed system.

### A. *Inter-Model Characterization or Qualitative Assessment*

Considering existing approaches applying failure recovery approaches towards fault-tolerant SDN, we explored and assessed performance of a few recent methods. Similar to the proposed HSPC-SDN model,authors in [22] performed continuous network monitoring and exploited network parameters to detect link-outage in the data plane without applying any sophisticated controller. Here, authors applied delay information to classify a device as malfunctions or faulty; yet could not address resulting recovery strategy. On the contrary, our proposed HSPC-SDN model focused on ensuring QoS/QoE demands to ensure delay-resilient proactive routing control (monitoring, fault identification and recovery path switching). Authors didn't quantify their performance in terms of QoS parameters. Unlike [22], authors in [24]proposed proactive failure recovery concept by applying dynamic network information and alternate path formation by means of a dedicated SDN controller unit. A similar approach was developed in [26] which exploited network's backup information processed with a heuristic named ant

colony system to identify forwarding path. Noticeably, this controller mainly focused on single forwarding path formation and didn't address dynamic link-outage problem, which is a common and most frequent disruption in contemporary SDB-based systems like IoT/M2M, WLAN, VASNET, VLAN etc.Moreover, authors applied shortest path information to perform forwarding path decision. A proactive failure recovery concept was designed for SDN-based systems in [27] as well. However, authors didn't focus more on QoS rather normal smart grid data delivery. Despite failure path recovery-based fault-tolerant routing in SDN, the contributions made in [41]lacked real-time problem address and can be confined while addressing congestion, link-dynamism, delay constraints as the real-time network demands [48]. Though, authors in [50] considered traffic congestion measurement to perform routing over SDN based urban communication systems; it could not address other QoS parameters including energy constraints,especially to be employed over SDN-based IoT/M2M purposes or even VASNET. An interesting approach was designed in [57], where authors hypothesized that a reliable path can be the only one having higher number of shared components. Yet, this approach applied single forwarding path as the best suitable solution and didn't address future failure probability due to link-outage,malicious attacks or even hardware failure. Similar to our proposed HSPC-SDN model, authors in [62] applied a heuristic concept named breadth first search to perform failure recovery in SDN control. Yet, authors failed in addressing QoS-sensitive routing or allied decision making to guarantee fault-tolerant communication. To alleviate aforesaid problem, authors in [63] advocated non-overlapping path estimation and backup path formation for failure recovery task. Yet, as an illustration,authors applied merely Dijkstra and Disjoint Path (DP) algorithm for backup path estimation towards SDB-based system.

Observing above discussed key recent works and allied strengths as well as weaknesses, it can easily be found that the proposed HSPC-SDN model addresses major at hand limitations (of the existing methods, as discussed above), whether in terms of proactive failure recovery strategy, delay resilient self-configuring failure recovery ability, or heuristic driven multi-path disjoint (failure recovery) recovery concept. Undeniably, the inclusion of multi-parametric risk assessment at first strengthens the proposed HSPC-SDN model to thwart away any possible failure in future. Since, the node information collected towards risk assessment (in NIB)is applied for heuristic driven path disjoint path selection as well, it reduces any additional computational overheads and make proposed model more time-efficient as well as cost (say, energy)-efficient. On the other hand,unlike other recovery path estimation approaches, as discussed above our proposed HSPC-SDN model applied logical-control assisted self-configuring ability to implement failure recovery, and therefore reduces delay as typically found in existing methods employing node discovery, fault-detection, forwarding path estimation and recovery path switching delay. It confirms that the ability to process heuristic driven proactive (self-configuring) recovery path assignment makes the proposed HSPC-SDN model more efficient. This as a result can help accomplishing QoS/QoE demands. To be noted, despite being close to our proposed HSPC-SDN model, almost all existing approaches including [22] [24] [26] [27] [42] [48] [50] [57] [62] [63]have measured performance qualitatively [22] [41] or with certain specific performance metrics like delay [26] [57] [62] [66], link utilization [47], packet loss [23] [26] and recovery time [63]. A recent work as discussed in [62] made effort to improve SDN scalability and fault-tolerance where it focused on estimating the number of possible paths in between the controller and the switches to improve convergence time and data transmission rate. Similar to our approach, it applied MAC information (MAC ID) and intended to reduce hops in path to support reliability. Yet, it failed in addressing major adversaries including link-outage due to topological charges, hardware loss, malicious attacks etc. Exploring in depth, the involved computational exhaustion especially in Hierarchical labeling, network (root) discovery, fault labeling, forwarding logic and reconfiguration policies make this approach highly exhaustive. Authors in [66] developed dynamic disjoint path-based network failure recovery concept for SDBbased IoT ecosystems, where they considered wired,wireless and hybrid media to assess their performance.Despite efficacy, authors failed in addressing risk assessment before forwarding path estimation, which could have made it more robust to alleviate iterative link-failure probability.

### *B. Quantitative Assessment*

Noticeably, above discussed approaches have merely addressed limited performance goal or quantifiable performance parameters. On the contrary, in this paper we focused on accomplishing a solution which could guarantee QoS provision in terms of high packet delivery rate (PDR%), low packet loss (PLR%), negligible delay (sec) and energy consumption. Considering this fact, we identified a recent literature discussing a fault-tolerant and QoS-oriented SDN control model for IoT driven system [66]. Though, authors have developed their proposed Shared Risk Link Group (SLRG) based model for SDB-based IoT systems in Mininet, we redesigned their specific contribution of hybrid disjoint path formulation-based failure recovery concept. Noticeably, though authors [66]had contributed different modalities including Disjoint Path for wired, wireless and hybrid network, we considered only wireless disjoint path formation concept with single link-failure. Similar to our approach, the method proposed in [64]estimates the back-up (i.e., recovery) path at the start of network in addition to the in-run estimation. Though, our proposed model advocates multiple disjoint path estimation at the time of network initialization itself and therefore it doesn't require run-time path re-estimation. Unlike the proposed HSPC-SDN model, their proposed SRLG based disjoint path applies single recovery path and therefore to cope up with multiple link-failure it requires run-time path estimation. Summarily, the existing (here, reference model) SRLG-DD (wireless chan-

nel) approach considered dynamic disjoint path estimation concept [66]over wireless network where it obtained two disjoint paths one at the start of the network and the second is applied during run-time to cope-up with the link-failure and resulting recovery path demands. However, the existing SRLG-DD (wireless) model didn't consider risk analysis at the start of path formation that can make it vulnerable during run time. This as a result can make it confined and limited to cope up with network failure caused due to sudden link-outage, hardware malfunction, or hardware damage. This is because such damages might give rise to the iterative or one after another link outage and hence adopting such recovery demands with single backup path is not feasible. On the other hand,executing SRLG-DD iteratively can impose computational overheads as well as resource exhaustion and delay, thus can impact overall QoS performance. On the contrary, the ability to perform risk assessment prior to forwarding path estimation makes HSPC-SDN robust to alleviate future adversaries (especially due to misbehaving nodes, malfunction or malicious attacks). In addition, the use of self-configuring multiple forwarding paths (here, we deployed three backup paths, which are stored in NIB proactive table) makes HSPC-SDN robust to accommodate any failure recovery in run-time without undergoing any run-time path discovery processes. It makes proposed model more robust to meet QoS centric communication in SDN-based systems. The aforesaid efficacy confirms robustness and superiority of our proposed HSPC-SDN model over the existing SRLGDD model [66]. To quantify the relative performance of the proposed HSPC-SDN model and the existing SRLGDD model, we simulated both concepts and examined performance outputs in terms of PDR (%), PLR (%),recovery delay (ms), and energy-consumption (mj). The details of the performance metrics can be found in [65].HSPC-SDN model was designed using Network Simulator -2 software tool. The algorithms involved were developed using Object oriented programming language (OOPS), while the simulation was made over Ubuntu 14 operating system with CPU armored with 8 GB RAM. To assess scalability and overall efficacy of the proposed system, we simulated HSPC-SDN based model over different topologies, encompassing different switch densities or network sizes. The simulation conditions considered in this work are given in Table I. As depicted in Table I, in sync with LLN or WSN driven SDN based IoT as the application environment, we considered IEEE 802.15.4 MAC and PHY setup.

The simulation results obtained for both SRLG-DD and HSPC-SDN models are given in Fig. 3 to Fig. 6. In sync with QoS performance, SDN-based system requires to deliver high PDR (%), low PLR (%), minimum delay (ms) and energy consumption(mJ). To be noted, both SRLG-DD and HSPC-SDN models focus on achieving QoS performance. Looking into the simulation outputs (Fig. 3) it can easily be observed that the proposed HSPCSDN model exhibits average PDR of 96.21%. On the contrary, SRLG-DD method achieves PDR of 90.20%, which is significantly

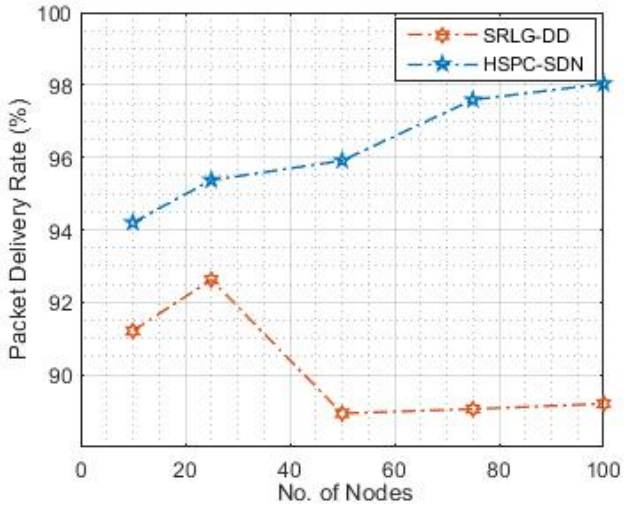


Figure 3. **Packet Delivery Ratio (PDR) Performance**

lower than the proposed HSPCSDN method. Fig. 4 presents PLR performance by the proposed HSPC-SDN and SRLG-DD. As depicted in the results, it can easily be observed that similar to the PDR performance, the proposed HSPC-SDN model exhibits significantly lower PLR in comparison to existing SRLGDD approach. Noticeably, the average PLR by HSPCSDN is 3.78%, while SRLG-DD exhibits the PLR of 9.61%, which is significantly higher than the proposed method. This can be mainly because of risk-aware routing decisions by the proposed system followed by its robustness to inculcate dynamic disjoint multiple path selection. To be noted, despite the fact that SRLG-DD applied disjoint path selection; yet under multiple failure condition it merely applied two disjoint paths, where identifying link-outage, it switched to the alternate path.

As depicted in the results (Fig. 3 and Fig. 4), with increase in network topology or network size, likelihood of outage increases and hence a network might undergo link-outage over dynamic link condition. In this reference, identifying the best forwarding route requires processing a large search space and its dynamic statistics. In this case, the method might undergo high packet drop (till the recovery path is executed) and delay (Fig. 5). In sync with this fact, since our proposed HSPC-SDN model performs risk assessment and identifies the set of most reliable nodes or devices only for forwarding decision, not only it reduces the search space but also makes routing more efficient.

In fault-tolerant routing approaches or SDN control mechanism, especially in link-failure recovery-based approaches, the delay is more important. There are numerous applications including SDN-based IoT, M2M etc. where ensuring delay-resilient transmission is inevitable to meet QoS and QoE demands. In sync with this motive, we examined recovery time analysis for both proposed HSPC-SDN and SRLG-DD methods.

TABLE I. **Experimental Setup**

| Parameter | Value |
|---|---|
| Number of Nodes or Devices Per Topology | 10,25,50,75,100 |
| Network Dimension | 100 * 100 |
| MAC | IEEE 802.15.4 MAC |
| PHY | IEEE 802.15.4 PHY |
| Radio | 100 meters |
| Transmission Rate (BPS) | 10-512 p/s |
| Career Frequency | 2.5 GHz |
| Antenna | Omnidirectional |
| Link Margin | 45dB |
| Gain Factor | 35dB |
| Power Density of Radio Channel | -130 dB m/Hz |
| Noise of the Receiver | 10dB |
| BER Performance | $10^{-2}$ |
| Channel Distribution | Constant |
| Power Consumption at the transmitter | 98.2 mill watts |
| Packet Size | 512 Kb |
| Simulation Time | 200 sec |
| Traffic type | UDP |
| GA Population Size | 10 |
| Number of Runs | Generations(10) |
| Fitness Value | Link-connectivity, number of hops, and path with no common component. |

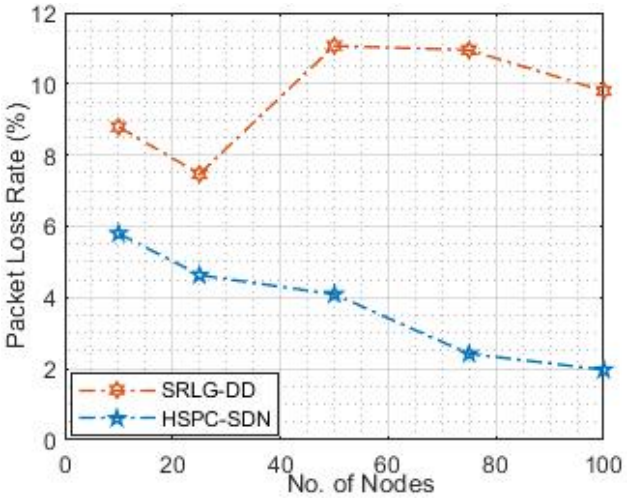


Figure 4. **Packet Loss Ratio (PLR) Performance**

Undeniably, both HSPC-SDN as well as SRLG-DD methods intended to reduce delay in recovery by applying backup recovery path; yet unlike our proposed method, SRLG-DD method performs path estimation at both network start as well as in run-time condition, where the later process is executed once identifying any link-outage. On the contrary, the proposed HSPC-SDN method performs multiple (here, three) disjoint path estimation and stores them in NIB. These disjoint paths are executed without undergoing iterative route discovery and selects recovery paths automatically by executing logical AND function. This approach helped the proposed HSPC-SDN achieving speedy recovery than the existing SRLG-DD method. Since, the delay cost increases as per the time consumed during forwarding path estimation, fault-detection, recovery path estimation and reconfiguration. In reference to this, since our proposed HSPC-SDN model once identifies any link-outage it executes logical AND function that helps selecting the alternate path from the NIB proactive table directly, without undergoing run-time recovery path estimation cost. This as a result makes HSPC-SDN more time efficient. On the contrary, SRLG-DD method requires undergoing online recovery path estimation that imposes delay, which is depicted in Fig. 5. As stated in Fig. 5, the proposed HSPC-SDN model takes significantly lower recovery time than the existing SRLG-DD method. Statistically, HSPC-SDN method takes 1.66 ms time, while the existing SRLG-DD method consumes a total of 11.13 ms recovery time. It shows that the proposed method can be more time-efficient to meet QoS/QoE demands in real-time systems. Since, the proposed system shows very less recovery time even over the large node density or large topology, it signifies scalability capability of this system.

Contemporarily, majority of the communication systems are battery-operated solutions that can also be called as the resource-constrained network. In such networks, including SDN-based IoTs or M2M communication systems guaranteeing minimum energy can help ensuring higher lifetime and longevity of the network. Moreover, since

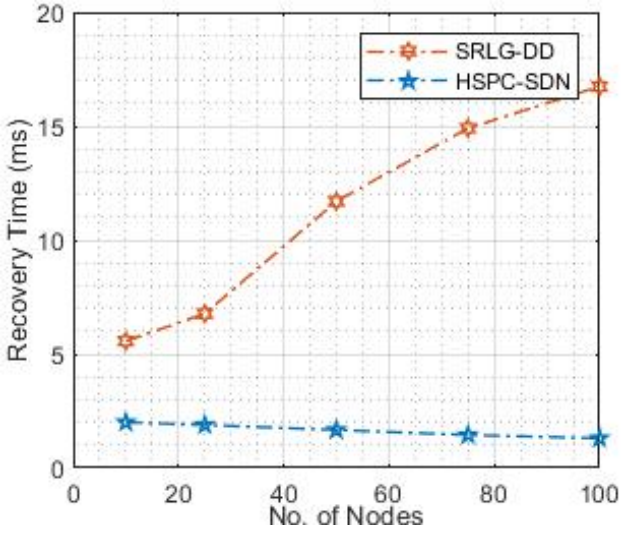


Figure 5. **Failure Recovery Time (ms) Performance**

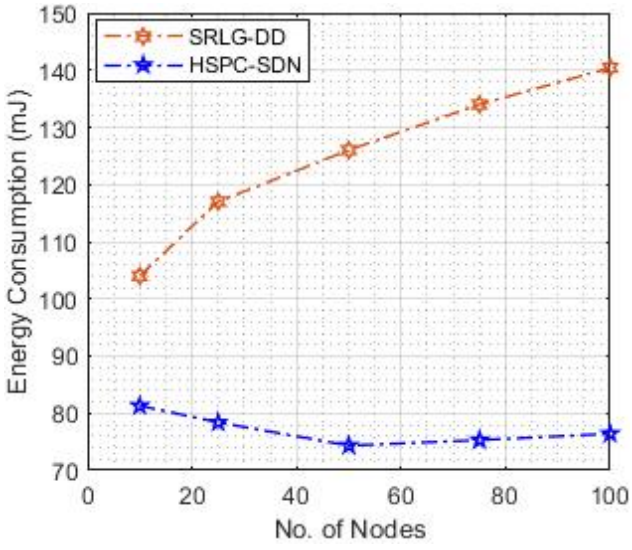


Figure 6. **Energy Consumption (mJ) Performance**

the energy consumption is directly related to the communication efficacy (i.e., PDR and PLR), maintaining low energy consumption signifies efficacy of the successful transmission. Considering this fact, we assessed the energy consumption which can be derived as the addition of the energy consumed during network deployment, NIB formation, proactive disjoint path estimation and reconfiguration. Statistically, the simulation results revealed that the proposed HSPC-SDN model consumes significantly lower energy (average 77.15 mJ) in comparison to the existing SRLG-DD method (average 124.28 mJ) (Fig. 6). The key reason behind higher energy exhaustion in SRLG-DD can be the higher PLR (%) as depicted in Fig. 4. Thus, taking into consideration of the overall performance, it can be inferred that the proposed HSPC-SDN model is more efficient than the existing approach (SRLG-DD). Moreover, the ability to ensure reliable transmission and QoS/QoE centric performance enables HSPC-SDN model efficient to serve major SDN-based network solutions. Observing overall performance outcomes and allied inferences, it can be stated that the proposed HSPC-SDN model agrees in affirmation with hypotheses defined (H01, H02 and H03). In reference to H01, unlike SRLG-DD method which don't consider any sophisticated risk assessment model before forwarding route decision, HSPC-SDN method performs rigorous multi-constraints risk assessment. This as a result helped guaranteeing higher PDR or successful transmission, as confirmed through Fig. 3 and Fig. 4. Therefore, the hypothesis H01 is found affirmative and hence accepted. The hypothesis H02 too confirms that the consideration of proactive multi-path failure recovery concept can enable fault-tolerant communication over SDN-based systems. Though, both HSPC-SDN as well as SRLG-DD applies proactive recovery concept, yet the ability to introduce automatic (proactive) recovery path selection makes the first more efficient. Thus, the hypothesis H02 is accepted positively. Similarly, despite the fact that both approaches apply disjoint path selection for data transmission; however, the provision of multiple disjoint paths while keeping higher network connectivity or availability with low distance makes proposed HSPC-SDN model more efficient than the SRLG-DD approach. The overall research conclusion and allied inferences are given in the subsequent section.

## 6. Conclusions and Future Work

In this paper a novel and robust Heuristic Driven Self-Configuring Proactive Controller is designed for QoS-centric SDN network (HSPC-SDN) was developed for SDN-based systems. The proposed HSPC-SDN model intended on addressing two key aspects, first to ensure that the data-plane controller considers only trustworthy and consistent nodes or devices for data communication, and second to introduce a disjoint multi-path failure recovery concept to guarantee delay-resilient transmission. In sync with the first goal, the use of dynamic network parameters including link-quality information, cumulative congestion degree, probability of successful transmission and link quality change index helped in segmenting the best and potential forwarding node (say, device) selection so as to thwart away any possible link-outage during run-time. The use of these key network parameters helped in retaining reliable nodes to support QoS-centric fault-tolerant transmission in SDN-based systems. Subsequently, this research hypothesized that reducing or eliminating any common forwarding node in recovery path can alleviate possible iterative link-failure. In this reference, the proposed HSPC-SDN model applied genetic algorithm, a well-known heuristic that exploited link-availability or connectivity information, preconditioned at the fact that the candidate path doesn't carry any common or shard component to complete routing. In this manner, GA exploited aforesaid link-availability, number of hops with no shared component as objective function, which is a NP-hard problem to identify the set of three best

forwarding cum recovery paths. Since, the overall proposed model was designed as a proactive routing solution with network information base (NIB) driven self-configuration, it alleviated any possibility of iterative network discovery and allied recovery path assignment cost. It helped improving delay performance. Moreover, the use of AND-logic function to select recovery path automatically not only improved delay performance but also reduced computational cost. The proposed HSPC-SDN model was designed in such manner that it can be applied as a standalone data-plane controller or as a routing solution or middleware of SDN based systems or devices. Simulation based performance assessment revealed that average packet delivery rate of 98.03%, packet loss rate of 1.97%, recovery time of 1.66 ms and energy consumption of 77.14 mJ over other disjoint forwarding path based SDN controllers. Relative performance assessment with existing QoS-oriented disjoint path selection based SDN controller revealed that the proposed HSPC-SDN model achieves average PDR of 96.21%., which is higher than the existing disjoint path based SDN control and recovery model named SRLG-DD (90.20%). Similarly, HSPC-SDN exhibited 3.78% of PLR which is significantly lower than the existing SRLG-DD (9.61%). The recovery time analysis confirmed that the proposed HSPC-SDN model takes merely 1.66 ms time, which is significantly lower than the existing method (SRLG-DD, 11.13 ms). The energy efficiency analysis too confirmed that the HSPC-SDN consumes significantly lower energy (average 77.15 mJ) in comparison to the existing SRLG-DD method (average 124.28 mJ). This can be due to superior fault-tolerance and proactive self-configuring multiple path selection. The ability of HSPC-SDN to perform automatic reconfiguration by employing proactively estimated multiple disjoint paths over the reliable devices or nodes strengthens it to exhibit high PDR, low PLR and hence low delay. The low PLR along with auto-configuration ability makes HSPC-PDR more energy-efficient. The robustness of the proposed HSPC-SDN model affirms its suitability for large SDN-based systems where it can guarantee fault-tolerance while preserving QoS-motives.

## References


[1] C. V. N. Index, "Global mobile data traffic forecast update, 2016–2021 white paper," *Cisco: San Jose, CA, USA*, vol. 7, p. 180, 2017.

[2] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "Sdmec: Software defined system for mobile edge computing," in *2016 IEEE international conference on cloud engineering workshop (IC2EW)*. IEEE, 2016, pp. 88–93.

[3] D. Suh, S. Jang, S. Han, S. Pack, M.-S. Kim, T. Kim, and C.-G. Lim, "Toward highly available and scalable software defined networks for service providers," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 100–107, 2017.

[4] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, "Quality of service terminology in ip networks," *IEEE communications magazine*, vol. 41, no. 3, pp. 153–159, 2003.

[5] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications surveys & tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[6] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[7] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[8] T. Bakhshi, "State of the art and recent research advances in software defined networking," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[9] O. Blial, M. Ben Mamoun, and R. Benaini, "An overview on sdn architectures with multiple controllers," *Journal of Computer Networks and Communications*, vol. 2016, 2016.

[10] A. Malik, B. Aziz, A. Al-Haj, and M. Adda, "Software-defined networks: a walkthrough guide from occurrence to data plane fault tolerance," PeerJ Preprints, Tech. Rep., 2019.

[11] W. Braun and M. Menth, "Software-defined networking using openflow: Protocols, applications and architectural design choices," *Future Internet*, vol. 6, no. 2, pp. 302–336, 2014.

[12] J. Chen, J. Chen, F. Xu, M. Yin, and W. Zhang, "When software defined networks meet fault tolerance: a survey," in *International conference on algorithms and architectures for parallel processing*. Springer, 2015, pp. 351–368.

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

[14] M. Paliwal, D. Shrimankar, and O. Tembhurne, "Controllers in sdn: A review report," *IEEE access*, vol. 6, pp. 36 256–36 270, 2018.

[15] "Openflow networking summit," http://opennetsummit.org/, apr. 2012.

[16] M. Desai and T. Nandagopal, "Coping with link failures in centralized control plane architectures," in *2010 Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010)*. IEEE, 2010, pp. 1–10.

[17] A. Malik, B. Aziz, M. Adda, and C.-H. Ke, "Smart routing: Towards proactive fault handling of software-defined networks," *Computer Networks*, vol. 170, p. 107104, 2020.

[18] M.-L. Chiang, H.-C. Hsieh, and C.-W. Wang, "Improving the fault-tolerance under software-defined network based on new sight of agreement protocol," *IEEE Access*, vol. 6, pp. 40 898–40 908, 2018.

[19] Y. Yu, X. Li, X. Leng, L. Song, K. Bu, Y. Chen, J. Yang, L. Zhang, K. Cheng, and X. Xiao, "Fault management in software-defined networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 349–392, 2018.

[20] A. M. Al-Sadi, A. Al-Sherbaz, J. Xue, and S. Turner, "Routing algorithm optimization for software defined network wan," in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*. IEEE, 2016, pp. 1–6.

[21] S. M. Raza, S. Ahvar, R. Amin, and M. Hussain, "Reliability aware multiple path installation in software-defined networking," *Electronics*, vol. 10, no. 22, p. 2820, 2021.

[22] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takács, and P. Sköldström, "Scalable fault management for openflow," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 6606–6610.

[23] R. M. Ramos, M. Martinello, and C. E. Rothenberg, "Slickflow: Resilient source routing in data center networks unlocked by openflow," in *38Th annual IEEE conference on local computer networks*. IEEE, 2013, pp. 606–613.

[24] ——, "Data center fault-tolerant routing and forwarding: An approach based on encoded paths," in *2013 Sixth Latin-American Symposium on Dependable Computing*. IEEE, 2013, pp. 104–113.

[25] N. Beheshti and Y. Zhang, "Fast failover for control traffic in software-defined networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 2665–2670.

[26] S. Zhu and S. Lan, "Action based proactive node failure protection mechanism for openflow," in *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*. IEEE, 2015, pp. 65–70.

[27] D. Gyllstrom, N. Braga, and J. Kurose, "Recovery from link failures in a smart grid communication network using openflow," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2014, pp. 254–259.

[28] F. Al-Tam and N. Correia, "Fractional switch migration in multi-controller software-defined networking," *Computer Networks*, vol. 157, pp. 1–10, 2019.

[29] N. Correia and A.-T. Faroq, "Flow setup aware controller placement in distributed software-defined networking," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5096–5099, 2019.

[30] M. Reitblatt, M. Canini, A. Guha, and N. Foster, "Fattire: Declarative fault tolerance for software-defined networks," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 109–114.

[31] N. E. Petroulakis, G. Spanoudakis, and I. G. Askoxylakis, "Fault tolerance using an sdn pattern framework," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[32] C. Cascone, D. Sanvito, L. Pollini, A. Capone, and B. Sanso, "Fast failure detection and recovery in sdn with stateful data plane," *International Journal of Network Management*, vol. 27, no. 2, p. e1957, 2017.

[33] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Openflow: Meeting carrier-grade recovery requirements," *Computer Communications*, vol. 36, no. 6, pp. 656–665, 2013.

[34] ——, "Fast failure recovery for in-band openflow networks," in *2013 9th international conference on the Design of reliable communication networks (DRCN)*. IEEE, 2013, pp. 52–59.

[35] M. Borokhovich, L. Schiff, and S. Schmid, "Provable data plane connectivity with local fast failover: Introducing openflow graph algorithms," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 121–126.

[36] N. L. Van Adrichem, B. J. Van Asten, and F. A. Kuipers, "Fast recovery in software-defined networks," in *2014 Third European Workshop on Software Defined Networks*. IEEE, 2014, pp. 61–66.

[37] D. Katz and D. Ward, "Bidirectional forwarding detection (bfd)," Tech. Rep., 2010.

[38] T. Pfeiffenberger, J. L. Du, P. B. Arruda, and A. Anzaloni, "Reliable and flexible communications for power systems: Fault-tolerant multicast with sdn/openflow," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2015, pp. 1–6.

[39] P. Thorat, S. M. Raza, D. S. Kim, and H. Choo, "Rapid recovery from link failures in software-defined networks," *Journal of Communications and Networks*, vol. 19, no. 6, pp. 648–665, 2017.

[40] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Enabling fast failure recovery in openflow networks," in *2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2011, pp. 164–171.

[41] H. Kim, M. Schlansker, J. R. Santos, J. Tourrilhes, Y. Turner, and N. Feamster, "Coronet: Fault tolerance for software defined networks," in *2012 20th IEEE international conference on network protocols (ICNP)*. IEEE, 2012, pp. 1–2.

[42] S. Kim, J. Son, A. Talukder, and C. S. Hong, "Congestion prevention mechanism based on q-leaning for efficient routing in sdn," in *2016 International Conference on Information Networking (ICOIN)*. IEEE, 2016, pp. 124–128.

[43] K. Nguyen, Q. T. Minh, and S. Yamada, "A software-defined networking approach for disaster-resilient wans," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2013, pp. 1–5.

[44] J. Li, J. Hyun, J.-H. Yoo, S. Baik, and J. W.-K. Hong, "Scalable failover method for data center networks using openflow," in *2014 IEEE network operations and management symposium (NOMS)*. IEEE, 2014, pp. 1–6.

[45] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 3, pp. 105–110, 2008.

[46] K. Lee, M. Kim, H. Kim, H. S. Chwa, J. Lee, and I. Shin, "Fault-resilient real-time communication using software-defined networking," in *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2019, pp. 204–215.

[47] M. M. Tajiki, M. Shojafar, B. Akbari, S. Salsano, M. Conti, and M. Singhal, "Joint failure recovery, fault prevention, and energy-efficient resource management for real-time sfc in fog-supported sdn," *Computer Networks*, vol. 162, p. 106850, 2019.

[48] B. Yuan, H. Jin, D. Zou, L. T. Yang, and S. Yu, "A practical byzantine-based approach for faulty switch tolerance in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 825–839, 2018.

[49] S. Song, H. Park, B.-Y. Choi, T. Choi, and H. Zhu, "Control path management framework for enhancing software-defined network (sdn) reliability," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 302–316, 2017.

[50] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "Sdn-based real-time urban traffic analysis in vanet environment," *Computer Communications*, vol. 149, pp. 162–175, 2020.

[51] J. Bhatia, P. Kakadia, M. Bhavsar, and S. Tanwar, "Sdn-enabled network coding-based secure data dissemination in vanet environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6078–6087, 2019.

[52] Y. Lu and S. Zhu, "Sdn-based tcp congestion control in data center networks," in *2015 IEEE 34th international performance computing and communications conference (IPCCC)*. IEEE, 2015, pp. 1–7.

[53] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-assisted congestion control mechanism for 5g network using software-defined networking," in *2018 15th International symposium on wireless communication systems (ISWCS)*. IEEE, 2018, pp. 1–5.

[54] S. Güner, G. Gür, and F. Alagöz, "Proactive controller assignment schemes in sdn for fast recovery," in *2020 International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 136–141.

[55] R. K. Das, F. H. Pohrmen, A. K. Maji, and G. Saha, "Ft-sdn: a fault-tolerant distributed architecture for software defined network," *Wireless personal communications*, vol. 114, no. 2, pp. 1045–1066, 2020.

[56] A. Malik, R. de Fréin, and B. Aziz, "Rapid restoration techniques for software-defined networks," *Applied Sciences*, vol. 10, no. 10, p. 3411, 2020.

[57] A. Malik, B. Aziz, and M. Bader-El-Den, "Finding most reliable paths for software defined networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1309–1314.

[58] Z. Yang and K. L. Yeung, "Sdn candidate selection in hybrid ip/sdn networks for single link failure protection," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 312–321, 2020.

[59] M. Shojaee, M. Neves, and I. Haque, "Safeguard: Congestion and memory-aware failure recovery in sd-wan," in *2020 16th International Conference on Network and Service Management (CNSM)*. IEEE, 2020, pp. 1–7.

[60] S. L. Aljohani and M. J. Alenazi, "Mpresisdn: Multipath resilient routing scheme for sdn-enabled smart cities networks," *Applied Sciences*, vol. 11, no. 4, p. 1900, 2021.

[61] Y.-D. Lin, H.-Y. Teng, C.-R. Hsu, C.-C. Liao, and Y.-C. Lai, "Fast failover and switchover for link failures and congestion in software defined networks," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[62] D. Lopez-Pajares, J. Alvarez-Horcajo, E. Rojas, A. Asadujjaman, and I. Martinez-Yelmo, "Amaru: Plug&play resilient in-band control for sdn," *IEEE Access*, vol. 7, pp. 123 202–123 218, 2019.

[63] K. Bakhshi Kiadehi, A. M. Rahmani, and A. Sabbagh Molahosseini, "A fault-tolerant architecture for internet-of-things based on software-defined networks," *Telecommunication Systems*, vol. 77, no. 1, pp. 155–169, 2021.

[64] R. Mohammadi and R. Javidan, "Efsute: A novel efficient and survivable traffic engineering for software defined networks," *Journal of Reliable Intelligent Environments*, pp. 1–14, 2021.

[65] G. Li, D. Wang, T. Gallivan, and R. Doverspike, "On shared risk link group optimization," *Journal of Optical Communications and Networking*, vol. 4, no. 11, pp. B52–B57, 2012.

[66] K. B. Kiadehi, A. M. Rahmani, and A. S. Molahosseini, "Increasing fault tolerance of data plane on the internet of things using the software-defined networks," *PeerJ Computer Science*, vol. 7, p. e543, 2021.

**Mr Sharathkumar S** obtained his B.E Degree from C.I.T Gubbi, Karnataka, Master's (M.Tech) from RVCE, Bangalore. He has a teaching experience of 11 years and working as an Assistant Professor in the Dept. of I.S.E, Siddaganga Institute of Technology, Tumkur, Karnataka. At present, he is doing Ph.D in the Department of Computer Science and Engineering at Puducherry Technological University (erstwhile Pondicherry Engineering College). The area of interests include Software Defined Networks, Reliability, fault tolerance in SDN.

**Dr N Sreenath** is a Professor in the Department of Computer Science and Engineering at Puducherry technological University, (erstwhile Pondicherry Engineering College) Puducherry. He has 30 years of teaching experience. He did his B.Tech. in Electronics and Communication Engineering, JNTU College of Engineering, Ananthapur, M.Tech from University of Hyderabad and Ph.D from IIT Madras. His area of Interests include Optical Networks, WDM , High Speed Networks etc.