Integrating Resilient Tier N+1 Networks with Distributed Non-Recursive Cloud Model for Cyber-Physical Applications

Kennedy Chinedu Okafor[1,2*] and Omowunmi Mary Longe[2]
[1]Mechatronics Engineering, Federal University of Technology, Owerri, Nigeria
[e-mail: kennedy.okafor@futo.edu.ng]
[2]Electrical and Electronic Engineering Science, University of Johannesburg, South Africa.
[e-mail: omowunmil@uj.ac.za]
*Corresponding author: Kennedy Chinedu Okafor

Abstract

Cyber-physical systems (CPS) have been growing exponentially due to improved cloud-datacenter infrastructure-as-a-service (CDIaaS). Incremental expandability (scalability), Quality of Service (QoS) performance, and reliability are currently the automation focus on healthy Tier 4 CDIaaS. However, stable QoS is yet to be fully addressed in Cyber-physical data centers (CP-DCS). Also, balanced agility and flexibility for the application workloads need urgent attention. There is a need for a resilient and fault-tolerance scheme in terms of CPS routing service including Pod cluster reliability analytics that meets QoS requirements. Motivated by these concerns, our contributions are fourfold. First, a Distributed Non-Recursive Cloud Model (DNRCM) is proposed to support cyber-physical workloads for remote lab activities. Second, an efficient QoS stability model with Routh-Hurwitz criteria is established. Third, an evaluation of the CDIaaS DCN topology is validated for handling large-scale, traffic workloads. Network Function Virtualization (NFV) with Floodlight SDN controllers was adopted for the implementation of DNRCM with embedded rule-base in Open vSwitch engines. Fourth, QoS evaluation is carried out experimentally. Considering the non-recursive queuing delays with SDN isolation (logical), a lower queuing delay (19.65%) is observed. Without logical isolation, the average queuing delay is 80.34%. Without logical resource isolation, the fault tolerance yields 33.55%, while with logical isolation, it yields 66.44%. In terms of throughput, DNRCM, recursive BCube, and DCell offered 38.30%, 36.37%, and 25.53% respectively. Similarly, the DNRCM had an improved incremental scalability profile of 40.00%, while BCube and Recursive DCell had 33.33%, and 26.67% respectively. In terms of service availability, the DNRCM offered 52.10% compared with recursive BCube and DCell which yielded 34.72% and 13.18% respectively. The average delays obtained for DNRCM, recursive BCube, and DCell are 32.81%, 33.44%, and 33.75% respectively. Finally, workload utilization for DNRCM, recursive BCube, and DCell yielded 50.28%, 27.93%, and 21.79% respectively.

---

1. Introduction

Data Centers (DCs) are large-scale computational environments with redundant computing, power, data communication, temperature, and other security appliances in a dedicated space. One of the biggest challenges for most African Countries is how to link their physical networks (such as laboratories, markets, buildings, power/transport systems, among others) to a resilient cloud DC network. A resilient network is any system that incorporates redundant subsystems/components, facilities, etc, to eliminate sudden disruptions or outages. Clustering schemes, multiple application provisioning, and redundant backup storage, as well as power supplies, can facilitate failover. Priority application workloads needing minimal disruptions are first considered before others. This is particularly important in disaster planning and business continuity in DCNs.

With the global pandemic still ravaging the world, Universities need to quickly link their physical laboratories to the cloud for sustained academic activities. This will engage seamless access to laboratory networks running various application workloads. Resilient networks will increase tolerance and reduce downtimes using series and parallel reliability analytics. As a result, cost-effective disaster recovery and data replication systems will make the cloud more useful to organizations. The huge capacity to cut costs, increase agility and minimize risks has made CDIaaS a viable option.

However, the inability to infuse resiliency in cloud-based implementations can lead to unforeseen CPS risks and consequent service disconnections. To set up CDIaaS, there are minimum requirements such as single or multi-tenant network operation centers (NOC) [1]. Standardization/consolidation, virtualization, automation, and security are common projects in a DC transformation initiative [2]. In [3], modern computing applications rely on stable DC deployments to guarantee optimal operations. Google, Facebook, Amazon, etc are typical examples. For these organizations, an average failure on the cloud infrastructure is estimated as USD100,000/hr while failure in workload application is estimated at USD 500,000 – 1,000,000,000/hr.

As a solution, Cloud-based cyber-physical systems (CCPS) explore intelligent control algorithms and mechanisms to monitor systems processes in DC-NOCs. The hardware and tiny software subsystems in CPS are used for gathering various geospatial/temporal datasets for predictive analytics [4], [5]. With resiliency in place, CCPSs can coordinate interactions among physical systems in context-dependent ways sitting on the Cloud. To achieve large-scale results, CCPS applications use the computational strength of NOCs to investigate QoS stability. At a lower level, embedded systems are used to achieve the edge-to-cloud computational requirements where physical elements handle process control and report to the Cloud NOCs [6]. Examples of CPS that can be hosted on the Cloud NOCs include Smart-grid, advanced Cloud meters, automated vehicles, telemedicine-health systems, and the Industrial Internet of Things (IIoT)/automation pyramid, among others [7]. These examples of CPS offer spectacular benefits because of sensor-based communication capabilities with the Cloud. This requires large-scale computation with QoS economics on the Cloud. For instance, many wireless sensor networks/IoT can use the Cloud NOC for huge computation. A good example is found at the distributed robot garden for the processing of farm produce [8] and Mechatronics Engineering Laboratory, Federal University of Technology, Owerri, Nigeria. Cybernetics, mechatronics actuation, and process controls all need stable Cloud integrations. In this case, balanced agility and flexibility for the application workloads need urgent attention. There is a need to manage interconnectivity among nodes and machines using a resilient and fault-tolerance scheme in terms of CPS routing service.

Again, with cyber-physical geo-spatial sensing involving IoT sensor nodes, the Cloud NOC can be used to provide navigation, manipulation, and wireless coordination across parts of the CPS. Most Cloud-based resilient control systems have their focus on the intelligent control system (ICS) that pervades critical NOC infrastructure using federated learning [9]. This type of next-generation design considers system resilience. This is usually very difficult to quantify in legacy DC-NOC models, especially in the areas such as Cybersecurity, Machine-human interaction, and IoT complex scenarios [10].

To achieve seamless resilient deployment, there are Tier 1 to 4 (i.e., Tier N+1) DCs defined by standardized regulators/authorities [11]. This involves determining the CCPS-DC uptime needed for computing the CPS-NOC performance, as well as the QoS. The most basic DC consideration is a Tier 1 model used by small-scale organizations. This type has non-redundant capability such as single uplink including fewer servers. Tier 2 is like Tier 1 but has more active components with significant redundancy. Tier 3 is also similar but combines Tier 1 and Tier 2 capabilities in a more advanced form. While Tier 4 encompasses all the previous Tiers with higher configuration components. It is fully faulted tolerant as every subsystem is powered by various power sources as well as supporting full redundancy. The Tier 4 DCs are the most reliable and less prone to failure, especially for CPS applications. These Tier 4 DCNs are designed to house mission-critical compute systems.

In terms of CCPS availability involving DCs Tiers, the four identified levels needed for workload integration include Tier 1 with 99.671 percent availability (guaranteed); Tier 2 providing a 99.741% uptime guarantee; Tier 3 having 99.982 percent availability (guaranteed) and Tier 4 with 99.995 percent availability (guaranteed) [12]. Most Tier 1 networks and ISPs are found in Tier 4 such as AT &T, Verizon Enterprise Solutions, Vodafone Carrier Services, China Telecom, Orange (OpenTransit), NTT Communications, including regional Tier 1 network [13].

With the volumetric flow of data streams in a full-duplex mode, a DC engine with full support for analytics will advance the growth of Cyber-Physical Cloud-based workloads. Scalability in DCs is fundamental for the performance and dependability of Tier 1 networks since it leverages settlement-free interconnections. To adapt to dynamic application workload requirements, most architectural topologies are yet to be made fully agile and reconfigurable. The network for CPS must connect on-demand to a large pool of resources while also providing reliable/stable link connectivity. Existing works on both datacentric server networks are studied in [14], [15], Hamiltonian-connected, DCell, BCube [15]-[17], HyperBcube, Flecube, Layered Scalable Data Center (LaScaDa) [18]. These use intelligent switches with the servers to forward packets. Other interconnect server-centric topologies used in DCNs by the scientific community to address the challenges include Flecube, Ficonn, HyperFlaNet, and DPillar [2], [19], among others.

Another class of DCN configuration commonly seen in setups is the intelligent switch-based server-centric model. This uses traffic routing to convey packets in the DCNs. These are found in VL2, Clos Network, Fat Tree, JellyFish, DOS, and Hypac [3]. Previous topologies scale often too quickly (growing exponentially in size) or transiently slowly, resulting in performance bottlenecks. Also, the major issue with Tier N+1 legacy networks is that it requires enormous computational resources that need to be serviced. For example, RAM/ROM management, buffer storage, I/O processing, and CPU utilization, among others. There are no readily available APIs for multiple I/O sensing, especially with IoT-powered devices. The QoS issues have not been fully addressed with DNRCM considering Cyber-physical applications. This is especially important for managing complex interconnectivity among nodes and

2260

Kennedy et al.: Integrating Resilient Tier N+1 Networks with Distributed
Non-Recursive Cloud Model for Cyber-Physical Applications

machines [17], [18].

Distributed Non-Recursive Cloud Model (DNRCM) is proposed to support the Cyber-physical integration of educational laboratory equipment for remote learning and lab activities. The DNRCM is a new Tier-5 cross-cube-based server-centric DC network derived from previous distributed Cloud computing network research [3]. It has decentralized and non-recursively defined features targeted at alleviating the upper switch's bandwidth bottleneck and increasing scalability in any type of Tier 1 to 4 network. The architecture groups edge nodes into recursive structure clusters and then connects them with a well-crafted pattern where the systems of nodes coordinate seamlessly while increasing connectivity.

Applying control-based load balancing will enhance DataStream traffic flows in DNRCM, especially during congestion. The dynamic load balancing with Ubuntu OS can give QoS provisioning benefits while using the SDN Floodlight controller to derive optimal system topology. In this regard, Mininet (practical virtual network executing a real kernel, switch, and App code on a native Cloud VM) CLI will be sued for custom configuration and deployment in a production setting. It is interfaced via the SDN controller IP addressv6 on port 6653. The DNRCM setup is then executed to gather data from the physical topology later in section 4.

The aim contribution of this paper is to replace the legacy data-centric designs with a robust DNRCM-topology that is stable and scalable for CCPS infrastructure. The discussed DNRCM-topology uses non-recursive hierarchical row-induced routing to route packets between CPS nodes and the CCPS sink. Furthermore, the proposed topology uses a small-node degree to interconnect many CPS nodes.
The highlighted contributions are as follows:

- Characterizations of a typical DCN for University application workloads.
- Modeling of Non-Recursive QoS link cluster maximization using OpenFlow DNRCM construction stability criterion.
- Reliability Analytics for in-built series and parallel pod cluster design.
- Mathematical derivations for DNRCM/DCCN linearization and local stability model.
- An architectural design of the DNRCM/DCCN CPS integration involving edge and cloud IaaS QOS resource provisioning environment.

In this paper, efforts are made to organize the paper as follows. Section 2 presents related works on various data center networks. Section 3 presents the classical Tier 1 practical DCN survey and the related issues. Section 4 presents the taxonomy of DNRCM. Section 5 presents the experimental analysis and Section 6 concludes the paper with future directions in Section 7.

2. Literature Survey

2.1. Related Research Efforts

Various research efforts within the DCN domain have sufficiently highlighted complex network attributes with little bearing on CPS. Both datacentric server-centric and switch-centric DCNs have been studied previously [3]-[18]. Distributed cloud computing data center (DCCN) was applied in smart green Energy Internet [3]. The authors in [20] focused on a disaggregated application-centric optical network (DACON) for DC infrastructures using hybrid optical switches. Their work solved the current challenges regarding bottleneck performance issues and poor resource utilization in recent server centric DCNs. The core

solution is based on software define network (SDN) orchestration domain. Also, authors in [21] highlighted new perspectives on DCN machine learning automation. Their work addressed concerns in the areas of workload forecasting, traffic flow control, traffic classification and scheduling, topology management, network state prediction, root cause analysis, and network security. The work in [22] used SDN technology to build a topology-aware routing scheme thereby relieving the bandwidth and processing overhead on controllers. In [23], the authors presented RT-HCN as an indexing scheme for mapping *R*-tree-based indexed DCNs. The work puts together storage and computes nodes using HCN overlay for server-centric data center topologies designs. In [24], the authors proposed a DCN referred to as High Scalability Data Center Network Architecture (HSDC). The design is derived from the hypercube network [25], [26] and it is constructed from mm-port switches and 2-port servers. A fault-tolerant routing algorithm is leveraged for executing any implementation of the topological properties. In [27], the authors proposed a fast diagnosis algorithm of complexity degree. Zhang et al [28] discussed an optical interconnection network architecture based on distributed optical switches. The model explores a two-dimensional torus topology based on 5*5 optical switches per node. In [29], the authors investigated a novel architecture that is compliant with cloud-based medical imaging requirements using Kubernetes. Efforts in [30] investigated a new optical DCN referred to as ROTOS. This is based on baseline reconfigurable optical top-of-rack (ToR) and fast optical switches. The work employed multiple transceivers (TRXs) with a wavelength selective switch (WSS) reprogrammed with an SDN control plane. In [31], the authors developed a DCN that leverages a high-performance and scalable traffic optimization strategy (HPSTOS). This depends on a hybrid scheme that leverages the benefits of both centralized and distributed mechanisms to enhance the efficiency of flow detection via sampling and flow-table recognition. The authors [32] focused on a multiobjective optimization problem for internet datacenters while using functional algorithms to generate feasible points via a scheme that supports feasibility preservation. The work used a multiobjective evolutionary algorithm (MOEA) to create Pareto optimal workload while achieving optimal scheduling, scalability, and feasibility. In [33], software-defined networking (SDN) and network function virtualization (NFV) have been seriously applied to enhance DCN functionality through the instrumentality of programmability and flexibility attributes at scale. The work [34] looked at DCN fault tolerance via the construction of completely independent spanning trees (CISTs) in server-centric DCNs called BCube connected crossbars (BCCC). This offers optimal network performance leveraging inexpensive commodity off-the-shelf switches and commodity servers. In [35], [36], the authors extensively discussed BCube architecture built for shipping-container-based modular server-centric DCN. While BCube provisions several bandwidth-intensive services via hastening their traffic patterns, its maintenance is overly complex considering the shipping-container attributes. The work [37] proposed MDCube as a high-performance connectivity construct used to expand BCube containers into a complex DCN [38]. A representative sample of DCN literatures were studied extensively such as DCell [16], FiConn [39], Green DCN [40], HCN-BCN [41], Hedera [42], Jellyfish [43], OSA [44], PCube [45], REWIRE[46], iCautz's topology [47], vL2-WIM [48], R-DCN [49], Spine-leaf DCN [50]. These works made excellent contributions in terms of expandability concerns, latency, and network resilience.

2.2. Summary of Research Gaps

The major established research gaps are highlighted below.

1. The absence of in-built series and parallel reliability analytics is needed for incremental expandability and high-bandwidth provisioning.
2. Non-inclusion of probability density functions for failure time distribution especially in Clos networks.
3. Absence of linearization and local stability model needed for degree-diameter optimal graphs. This will guarantee robustness in component failure and sustain a regular structure for packet-level forwarding technologies.
4. Lack of optimization design for high-wiring complexity, and weakly localized re-routing constructs for full-duplex connectivity across various layers.
5. Nondeployment of full interconnection interface intelligence on NOC nodes and layer-3 switches.

3. Tier-I Practical DCN Survey

In this section, perspectives on practical DCNs will be discussed to identify gaps. We conducted a macroscopic traffic pattern analysis on a server centric DCN. The limitation found in existing networks is that the designs either scale too quickly (the sizable exponential growth in size) or too slowly, resulting in unacceptable oscillatory performance. The parasitic traffic flow-loops in legacy networks consume bandwidth and increase data traffic. This results in packet losses and delayed data transmission, negatively impacting QoS stability.

3.1 Scenario-1: UNN Data Center Network

The paper used Cloud resilient enterprise approach to understudy the existing baseline resilient deployments in UNN Cloud DC. The four pillars include one. Assessments and valuations. 2. Planning and design. 3. Implementation and design. 4. Management and sustained integration. In this Section, a study on traffic transactions on the largest UNN DCN is studied in **Fig. 1a**. Initial reliability analysis of the data center has been reported in [3], [51]. **Fig. 1** shows the server-centric topology with the physical data center NOC. The server is powered with Linux-MikroTik-RouterOS. On the network board, the features configured include firewall, virtual private network (VPN) service, bandwidth shaping, and QoS settings. Over 50 Access points are distributed for traffic tunneling. Datacenter trends for daily and weekly throughputs were gathered from all of the devices in the various virtual local area network (VLAN) interfaces of the NoC in **Fig. 1**. Looking at the traffic analysis profiles from the NOC, DCell and BCube attributes are observed. The NOC is used to gather traffic statistics and patterns from all the incoming and outgoing sources. The observed traffic trend from the interface statistics plots appears less busty with undulating oscillations. This amounts to an unreliable and unpredictable state of the network with a low bandwidth scale. This observation makes the network less applicable and insufficient to serve CPS applications, especially with thousands of IoT nodes. This throughput plot shows low throughput output that never exceeded 20Mbits/sec which is not efficient for edge-to-cloud transactional applications in CCPS. Similarly, **Fig. 2a-d**, shows a traffic reliability trend for daily and weekly throughputs that are barely insufficient for large-scale computing.

In terms of private Cloud resilience in **Fig. 1**, the following issues were observed: 1.) it was difficult to identify the resilience supports; 2.) It is not feasible to determine downtime costs/hours; and 3.) evidence of tested operational resilience, risk concentration associated with sources of data-defacing, and extent of impacts are all absent. On a large scale, traffic patterns appear unpredictable. Additionally, there are uneven, large-scale, and sporadically transactional volumes because of workload dynamics. These constraints in terms of system reliability will affect CCPS DCN.
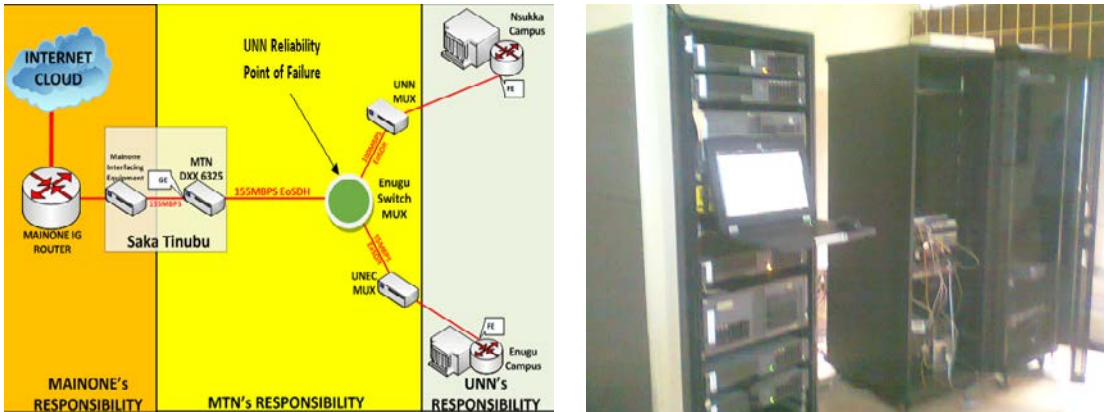

Fig. 1. UNN Data center topology for Enugu and UNN Campuses with NOC for Wi-fi Connectivity (Source: Authors Field survey with permission).
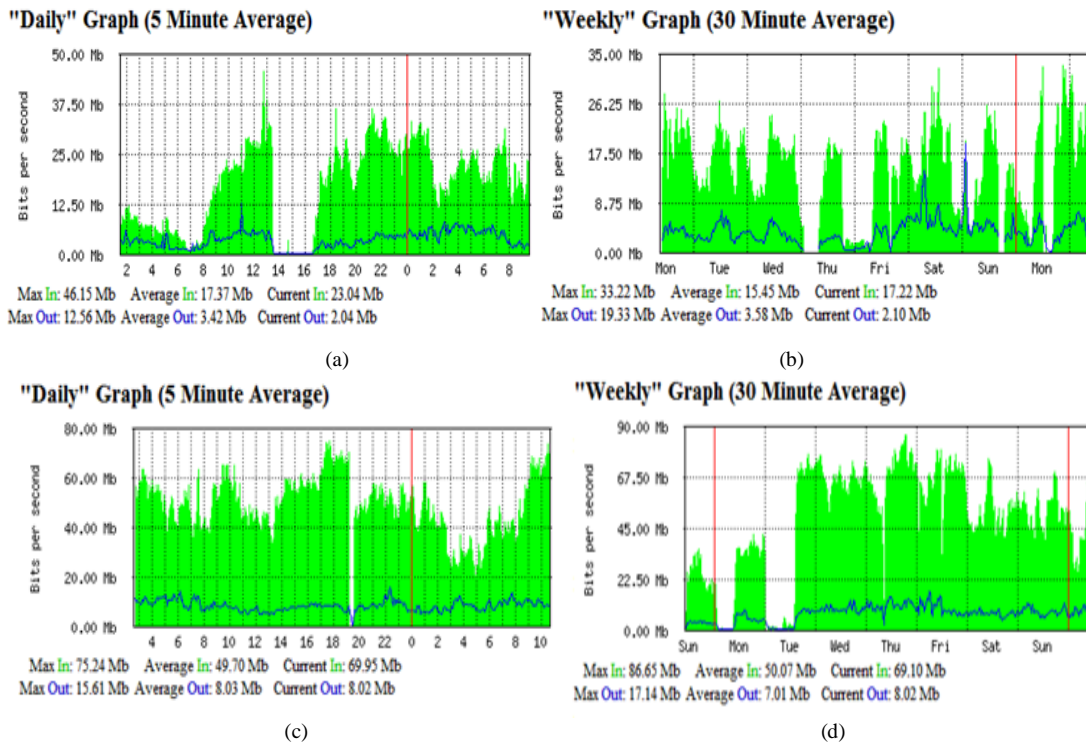

(a)

(b)

(c)

(d)

Fig. 2.a-d. Daily and Weekly Interface statistics plots on UNN DCN, 2020.

4. DNRCM Taxonomy

4.1. Building Blocks

Flexible Topology: DNRCM offers reliability, scalability, and incremental reconfigurability via its Pod clusters. Since smart DCNs offer cloud computing services at the layer of Infrastructure-as-a-Service (IaaS), incremental scalability is very key for CPS applications [52] running in DNRCM. As a result, the DCN infrastructure for CPS should be flexible and adaptive to meet the increasing CPS application workload requirements. The CCPS DCN must be carefully developed to maximize network bandwidth and deliver satisfactory QoS. In this Section, the DNRCM algorithm for the design topology in **Fig. 3** is shown. The system first absorbs the reliability requirement for QoS performance. The logical integration is then used for Edge and Fog connectivity concurrently**.**

Now, let the incremental expandability (i.e., scalability profile β) for DNRCM topology be derived from a model showing the precise node-count $\prod_i^{\infty} pN_c$ for pod clusters. This must be connected considering input specifications like number of ports per switch, number of layers, NL= $\sum_0^{\infty} N_l$. The scalability of the design topology can be easily realized using a nonrecursive incremental construction matrix. This is the concept of pod cluster expandability in DCNs. A pod cluster is defined as a group of $\forall$ servers linked to an external n-Port switch. It uses a cluster-driven fault-free-routing interconnection matrix (FFRIM) in a Pod while leveraging the Connection Failure Rate (CFR) metric for routable recognition and failure state condition map. When DNRM CFR slowly increments with the number of faulty components, then the network is elastic with reliable performance under faulty boundary conditions.

4.2. Structural Description

In design DNRCM, there are complex node-sets with various configuration parameters derived from a typical DCN design [18]. This normally comprises baseline DCN-nodes $(n_0 + n_1 + n_2 + n_3 \dots n_{j-1})$, DCN-switches $(s_0 + s_1 + s_2 + s_3 \dots s_{j-1})$, and DCN-links $(L_0 + L_1 + L_2 + L_3 \dots L_{j-1})$. The major links introduced in DCNs are illustrated in **Table 1** and these include i) links for two nodes ($\cup \rightarrow_{2n_{od}}$); 2) links for DCN node and switch ($\cup \rightarrow_{n_{od*switch}}$) and, 3) links for two switches (i.e., trunks) ($\cup \rightarrow_{2switches}$). The Alpha μ connection is assumed to most dependable DCN connection. The reason is that it supports several non-blocking routing paths for cascaded node to node or node to switch communication to the proposed DCN. Table I shows typical DCN topologies namely DCell [16], BCube [35], VL2 [53], FatTree [54], Ficonn [39]. Their respective scalability, bisection bandwidth, and diameter are highlighted in **Table 1**. The two variables given are = *Number of ports per node* ($k > 0$), n = *Number of ports per switch*($n > 0$) . From the table, FatTree and VL2 are seriously constrained in terms of scalability (zero scale-up). Only DCell and BCube seem to have better scalability although CPS deployments cannot be deployed on DCell and BCube. Unfortunately, high complexity in terms of wiring is the issue with DCell while BCube needs above 3-layers to scale up into a large-sized DCN. Consider a 4-Port-DCN-switch needed to design a DCN. This will imply deriving five layers for such a design., (i.e., $S = n^k = 4^5$=1024 nodes). Therefore, five interface network cards (NIC) will be required to build a 5-layer BCube DCN. This is not cost-effective. Besides, such a design can lead to worrisome wiring issues in CPS deployments. In Section 4.3, a discussion on the physical structure of the proposed CPS DCN is presented.

Table 1. DCN Topological Considerations.

DCN Features	Fat-Tree	VL2	DCell	FiConn	BCube
Scalability (S)	$\dfrac{n^3}{4}$	$5*n^2$	$\left(\left(n+\frac{1}{2}\right)^{2^k}-\frac{1}{2}(n+1)^{2^k}\text{-}1\right)$	$\geq \left(\left(\frac{n}{4}\right)^{2k}2^{k+2}\right)$	n^k
Bisection Bandwidth	$\dfrac{n^3}{8}$	$10n^2-20$	$\dfrac{S}{4log_n S}$	$> \dfrac{S}{4*2^k}$	$\dfrac{S}{2}$
Diameter	6	6	$> 2^k\text{-}1$	$> 2^k\text{-}1$	k

4.3. Physical Structure

Fig. 3 shows four clusters of the proposed DNRCM/DCCN pods introduced for CCPS deployment. In this work, a layered non-recursive topological model of DNRCM/DCCN is constructed out of n-port multilayer switches. The DNRCM/DCCN topology was constructed with basic n-port OpenFlow switches. To link up 128 nodes in DNRCM/DCCN topology, 32-internal and 32-external 4-port multilayer OpenFlow Multilayer switches are employed. This is achieved through the interconnection of layer-3 OpenFlow switch using n-port (i.e., internal switches). The layer-2 DNRCM/DCCN pods, also known as a cluster, is the first building block. A cluster is made up of n nodes that are connected to a single n-port switch as depicted in **Fig. 3**.

The OpenFlow Layer-1 DNRCM/DCCN switch is interconnected in a well-symmetric pattern that optimizes connectivity between clusters while eliminating redundant connections For ($k > 0$), a demonstration of how to build a layer-2 DNRCM/DCCN from the LaScaDa networks [18] is further discussed. A DCN-cluster is a collection of nth compute-servers linked by an "external" n-port Multilayer switch as shown in **Fig. 3**. The composite network parameters adapted in DNRCM/DCCN could be made higher than that of the legacy Clos, Fat Tree, BCube, DCell, and Jellyfish DCNs.

Recall that while the switch centric DCN model (VL2, Clos, FatTree, JellyFish, etc) explores intelligent switches for smart packet routing, the server-centric topologies (DCell, BCube, HyperBCube, Flecube, FiConn, etc) additionally forwards packet streams from the servers while leveraging the computational intelligent switches for layer 2/3 functions. This makes use of a recursive structure for node interconnection. The advantage of the proposed DNRCM/DCCN is the in-built reliability considerations for incremental scalability and QoS provisioning. We shall now look at in-built reliability constructs for series and parallel configurations of Pod clusters. This comprises the external switches, internal switches, and NOC servers in **Fig. 3**.

Fig. 4 Shows a typical DNRCM/DCCN topological integration (Tier-N+1 CPS). This fits into the legacy 3-Tier Internet service provider model (ISP). The complex Internet is segmented into an autonomous system (AS) that uses the Internet protocol (IP)v4/v6. It also uses a Border gateway protocol (BGP) routing scheme. This makes the extended DCNs interconnectivity feasible. The transport of Internet traffic is depicted in **Fig. 4** and we show how the ISPs are categorized into a 3-Tier model considering the various Internet service workloads. The backbone Internet provider supports traffic to existing ISPs only. The DCN components are provisioned for other Tier1-ISPs. The DNRCM is deployed at this layer for the exchange of Internet traffic with other DCN Tier providers.

All the Internet exchange points (IXPs) are connected by this layer to provision optimal QoS metrics via private settlement-free peering interconnections. The QoS is pushed via the backbones using private peering connections. Tier-1 ISPs usually own the DCN infrastructure and directly control traffic flows through its connections. Hence, Big data traffic volumes are processed for enterprise entities and users via the ASs. Finally, the Tier-2 ISP DCNs deliver Internet traffic to CPS-IoTs (end users) via Tier-3 ISPs which are usually national or regional providers. The major issue in **Fig. 4**, is the concern of reliability analytics and localization stability constraints for CPS connectivity via the Tier-2 layer.
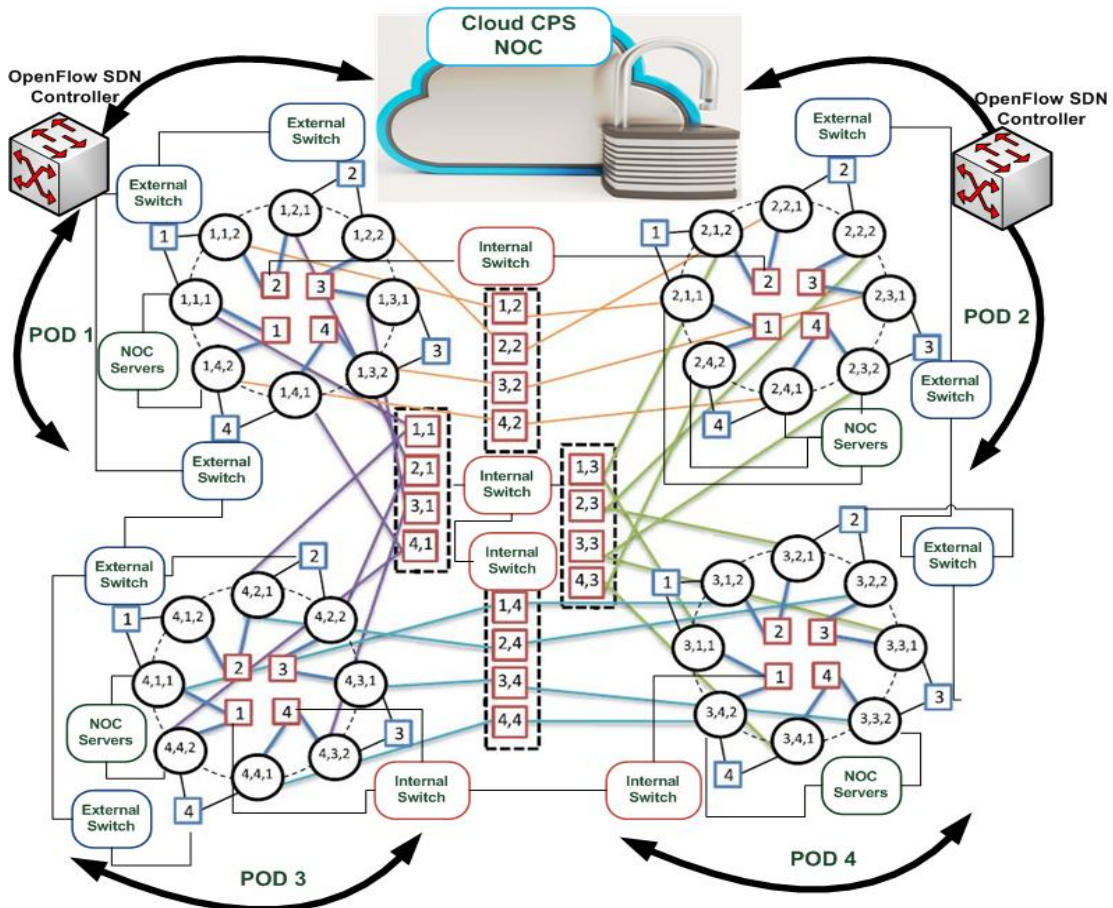

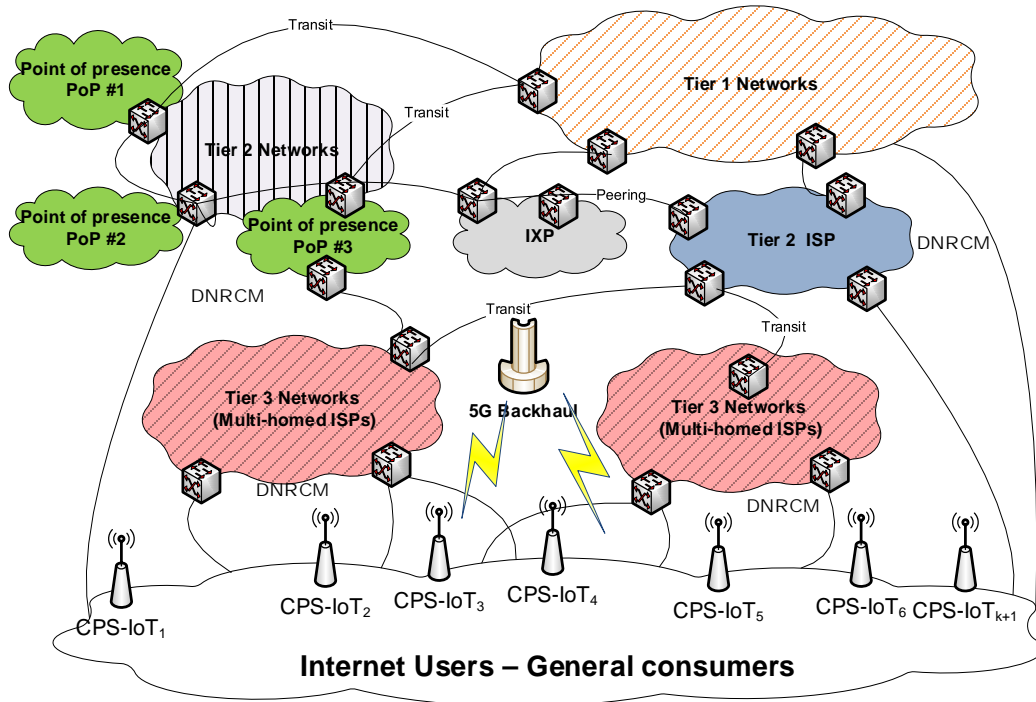**Fig. 3.** DNRCM/DCCN Topology for CPS Deployment.

Fig. 4. DNRCM/DCCN Topological integration (Tier N+1).

3.4. Reliability Analytics

To derive an appropriate model for DNCRM, we introduced Mean Time Between Critical Failure (MTBCF). This is used since hyper-scaled redundancy is configured in the proposed CPS NRDCN. For series configurations, let's consider Pod cluster γn components $\langle C_1, C_2, C_3, C_4 \ldots \ldots, and\ C_{n+1} \rangle$ arranged in cyclic series structure. The reliability of the Pod cluster series system (γ_{R_s}) assuming the cluster components to be isolated and independent, is given by (1):

$$\gamma_{R_s} = R_1 R_2 \cdots \cdots R_{n+1} = \prod_{i=1}^{n} R_i \tag{1}$$

Where R_i denotes the reliability of the Pod cluster series system (γ_{R_s}) component $C_i = C_{n+1}$. If the failure time of the ith Pod cluster component is given by $t_i, i = 1,2,3, \ldots, n,$ the in-built failure time of the series cluster system t_s is given by (2):

$$t_s = \min_{1 \leq i \leq n} t_i \tag{2}$$

Since a typical DCN pod fails as soon as one of its components fails. If $F_i(t)$ denotes the probability distribution function (PDF) of the failure time of the ith Pod component, the failure time distribution (FTD) of the series pod cluster is now given by (3):

$$F_s(t) = P(t_s \leq t) = 1 - P(t_s > t) = 1 - \prod_{i=1}^{n} P(t_i > t) = 1 - \prod_{i=1}^{n} [1 - F_i(t)] \tag{3}$$

Thus, the PDF of the failure time of the series pod cluster system is given by (4):

$$f_s(t) = \frac{dF_s(t)}{dt} = \sum_{j=1}^{n} \frac{\partial F_s}{\partial F_j} \frac{\partial F_j}{\partial t} = \sum_{j=1}^{n} F_j(t) \prod_{i=1}^{n} [1 - F_i(t)] \tag{4}$$

Now, let us determine the failure rate of the series pod cluster system. If the failure time of the ith Pod component follows an exponential distribution, with a constant failure rate λ_i (i = $1,2,\cdots,n$), then (5) holds.

$$R_i(t) = \prod_{i=1}^{n} [R_i(t)] = e^{-(\sum_{i=1}^{n} \lambda_i)t} = e^{-\lambda_s t} \tag{5}$$

Where the failure rate of the pod cluster λ_s is given by (6):

$$\lambda_s = \sum_{i=1}^{n} \lambda_i \tag{6}$$

So far, the Mean Time Between Critical Failure (MTBCF) for the series pod cluster is given by (7):

$$MTBCF(\mu) = \int_0^{\infty} R(t)dt = \int_0^{\infty} \left(\prod_{i=1}^{n} R_i(t) \right) dt \tag{7}$$

If the pod cluster failure rates of the components follow an exponential distribution, (7) now yields (8):

$$MTBCF(\mu) = \int_0^{\infty} e^{-(\sum_{i=1}^{n} \lambda_i)t} dt$$

$$MTBCF(\mu) = \left(-\frac{1}{\sum_{i=1}^{n} \lambda_i} \right) \left[e^{-\left(\frac{1}{\sum_{i=1}^{n} \lambda_i} \right)t} \right]_0^{\infty} = \frac{1}{\sum_{i=1}^{n} \lambda_i} \tag{8}$$

$$= \text{Total operational time (T)} \Big/ = \text{Total number of Failures (TN)}$$

Presentation in (8) accounts for pod cluster breakdown cost and failure frequency. Also, with (8), DCN inventory planning, capital expenditure (CAPEX) budgeting, and maintenance schedule automation, among others can be addressed.

Now, Since the external, internal switches, and NOC servers are connected in parallel at higher levels, the reliability of the pod cluster parallel clusters is given by the sum of the probabilities of realizing first i^{th} events. Hence, we have (9):

$$R_p = 1 - P_f = \prod_{i=1}^{n} (1 - R_i) \tag{9}$$

The relation in (9) can be generalized as follows. If Pod cluster γn components $\langle C_1, C_2, C_3, C_4 \ldots \ldots, and\ C_{n+1} \rangle$ arranged in a cyclic parallel structure, the reliability is then given by (10):

$$R_p = 1 - P_f$$

$$= 1 - P_f = \prod_{i=1}^{n} (1 - R_i) = 1 - R_p = 1 - \prod_{i=1}^{n} F_i \tag{10}$$

Where $F_i = (1 - R_i)$ denotes the probability of failure of the ith *the* component at time t. Now, the system failure rate $h_p(t)$ is given by (11):

$$h_p(t) = \frac{f(t)}{1 - F(t)} = \frac{\sum_{j=1}^{n} f_j(t) \prod_{i=1, i \neq j}^{n} F_i(t)}{1 - \prod_{i=1}^{n} F_i(t)} \tag{11}$$

Let's now look at the MTBCF of pod cluster parallel clusters. In this case, let the PDF of the ith the component at time t, be exponential with a failure rate of λ_i, i=1,2,3,4,······ n. The MTBCF is computed as (12):

$$MTBCF(\mu) = \int_0^\infty R(t)dt = \int_0^\infty \left[1 - \prod_{i=1}^{n} (1 - R_i(t)) \right] dt$$

$$= \int_0^\infty \left[1 - \prod_{i=1}^{n} \{ 1 - e^{-\lambda_i t} \} \right] dt$$

$$MTBCF(\mu) = \int_0^\infty \left[1 - (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t}) \cdots\cdots (1 - e^{-\lambda_n t}) \right] dt \tag{12}$$

Where $\lambda_n t$ is the critical reliability hazard function needed for availability and fault-tolerance. Presentation in (12) is the measure of Pod cluster Availability ψ (%) =

$$\frac{Uptime}{Uptime + downtime} \tag{13}$$

For pod cluster design, the NOC server availability is built on the reference NEC Express5800/ft (series servers) [55]. This provides 99.999% availability measured. It is opined that a well-managed DCN system can suffer from 5.25 minutes of downtime in a year. For 99.9999% availability, the annual downtime (ADTa) is 32 secs. For 99.999% availability, this should give 5 mins,15 secs (ADTa). For 99.99% availability, this gives 52 mins, 34 mins ADTa. For 99.9% availability, the (ADTa) is 8 hrs and 46 mins. Finally, for 99.9% availability, the (ADTa) 3 days, 15 hrs, 36 mins.

After obtaining the reliability requirement, the DNRCM/DCCN LAN/Optical components are deployed as detailed in [3], [6] with various test cases such as fault injection failure resilience (FIFR), security vulnerability protections (SVP), among others. The DNRCM/DCCN is used for external connectivity, especially in CPS-end devices pooling data from end-users. Considering Tier N+1 reliability considerations, all the fault-tolerant series-parallel DCN elements are arranged at higher levels. Redundant power supply, CPU, memory, I/O devices, cooling fans, etc, are applied to DNRCM/DCCN as additional supports. The idea is to maintain non-stop operations, non-disruptive maintenance, and operating system stability. In context, the network diameter N_d, number of nodes N_n, number of switches N_s, number of links N_l, and number of end-devices N_{ed}, were all captured in the topology.

To juxtapose the legacy data-centric network designs with a robust DNRCM-topology, we considered the reliability, stability, scalability, and cost-effectiveness of a modernized CCPS-data center networking infrastructure. A non-recursive algorithm that provides for end-to-end connectivity is presented. This considers all the various components of the DNRCM/DCCN as shown in **Table 1** and Algorithm I. The formulation for DNRCM-topology uses a hierarchical row-injection non-recursive routing scheme to route packets between CPS nodes. The proposed topology uses a small-node degree to interconnect a large number of CPS nodes. It also uses fault-tolerant routing to communicate to the clusters from source to destinations. The SDN OpenFlow hazard function is used to check for reliability link failure or any logical

and non-logical reliability issues. The inputs in Algorithm I bring the CPS nodes from various sources. It then maps the destination ports while scheduling the controls for the OpenFlow SDN controller. The QoS path is mapped by default while checking for all parameters of failure modes. The routing procedure uses fault tolerance to determine the longevity of the CPS nodes. Connectivity to the OpenFlow controller allows for massive convergence which impacts the performance of the network as depicted in the control structure of the algorithm. This is the part that enforces intelligence logically in the SDN centralized controller. This projects the entire network outlook to the applications and policy planes in the switch.

Algorithm I. Non-Recursive QoS link Cluster Maximization //OpenFlow DNRCM Construction Stability Criterion

1:	**Inputs:**	CPS-nodes; Source ( ), Destination ( ). CallSchedule
2:		History of CPS compute resources, QoS Provisioning, and transactional workflow
3:		/* K depicts the number of switch ports in DNRCM cluster.*/
		CPS Traffic-matrix CTM, CPS Network Topology (CNT), CPS Link Capacity (CLC)
4:	**Output:**	QoS Path is the metric path from the source to the destination, s;
		Dropping Max link-utilization routing allocation for data streams, d
5:	**Parameters:**	row-based routing ←Empty; // Fault Free Routing Scheme
6:		Connection failure rate ( )
		For CPS aggregate flows, F to F_{n+1}
		Enumerate all possible route paths I to in+1
		SDN_listing= Call Dijkstra shortest-path// locate various routes with small path length
		For Path j in CPS edge to SDN controller, do
		Create max. link-utilization of resources
		Allocate resources to links
		Update OpenFlow Switch table
		End
7:	**Procedure**	FaultTolerantRouting(QoS.MaxlifeTime)
8:		CPS-nodes. Hops refer to the No. of hops used in the Pod clusters
		Section I: /* construct DNRCMs */
		For each connected node $(C_k ; C_{k-1}........) \in C_{k-1}$
		Call SDN_OpenFlow ( )
9:	**Section I:**	/*create Non_recursive. DNRCM$_s$ nodes*/
		Invoke Hazard function ( );
		for (int i = 0; i < s; i++)
		Build Non_recursive.DNRCM$_s$ ([pref, *k*], s)
		Connect DNRCM $_s$ (s) to its OpenFlow switch;
		Check for reachability ( ); // Connection failure
		rate () for case of routing protocol not finding
		valid route.
		end for;
10:	**While** *i* < DNRCM s_Routing failed and CPS-nodes. Hops ( ) < MaxlifeTime do // monitorCallSchedule
11:		**Call** Hazard function ( );
		If reliability.hazard function ( ) is zero **then**
		Identify neighbor-severs in a radius.
		Call routing by replacing the selected node as a new Source.
		Select only Routes shorter than MaxlifeTime
		MaxLifeTime is maximum number of hops // for
		CPS-nodes. Hops
		CPS-nodes. Hops= 0
		Return
		construct Non_recursive.DNRCM$_s$ ([pref, *k]*, s)
		i ++;
		end if
12:	**end while**	
13:	CPS-nodes. Hops = CPS-nodes. Hops +1 ( )	
	Call CFR ( );	

> **If** fault free routing = false
> **end if**
> **If** fault-tolerant routing = false
> *end if*
> 14: **End**
> 15: **end procedure**

4.5. DNRCM/DCCN Linearization and Local Stability Model

Besides the series and parallel reliability characterizations established previously from (1) to (13), let's look at the uniform load balancing constructs. Now, the related mathematical models for the equilibrium points and stability criterion is discussed in Case 1 and Case 2 respectively.

Case 1:
Let us consider the DNRCM/DCCN model form in (14):

$$\frac{dx}{dt} = f(x) \tag{14}$$

whose local stability analysis is what we want to perform about the equilibrium point x^* (obtained by putting $f(x) = 0$).
We shall give a small perturbation to the system about the equilibrium point x^*.
Mathematically, this means we put $x = X + x^*$ into (14) to yield (15).
$\frac{dX}{dt} = f(x^* + X) = f(x^*) + Xf'(x^*) + \cdots$.(higher-order terms).
$\frac{dX}{dt} \approx f'(x^*)X$, since $f(x^*) = 0$ and neglecting higher-order terms.
Therefore, DNRCM/DCCN is stable if $f'(x^*) < 0$ (decreasing function) and unstable if $f'(x^*) > 0$ (decreasing function).
If $f'(x^*) = 0$, then DNRCM/DCCN linear stability remains inconclusive.

Case 2:
Let us now consider the model given by the system of differential equations of the form
$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y) \tag{15}$$

Let (x^*, y^*) be the steady-state solution of (15), then $f(x^*, y^*) = 0$ and $g(x^*, y^*) = 0$. We now give a small perturbation to the system about the steady-state, and mathematically this means we put $x = X + x^*$ and $y = Y + y^*$. This implies:

$$\frac{dX}{dt} = f(x^* + X, y^* + Y) \tag{16}$$

$\frac{dX}{dt} = f(x^*, y^*) + Xf_y(x^*, y^*) + \cdots$ higher-order terms by Taylor's series expansion of two variables. Similarly,
$\frac{dY}{dt} = g(x^*, y^*) + Xg_x(x^*, y^*) + Xg_x(x^*, y^*) + \cdots$ higher-order terms where
$f_x(x^*, y^*)$ is $\frac{\partial f}{\partial x}$ evaluated at the steady state (x^*, y^*). Since, by definition, $f(x^*, y^*) = 0$, and $g(x^*, y^*) = 0$, by neglecting second and higher-order terms, we get:

$$\frac{dX}{dt} = f_x(x^*, y^*)X + f_y(x^*, y^*)Y$$

$$\frac{dY}{dt} = g_x(x^*, y^*)X + g_y(x^*, y^*)Y \tag{17}$$

By putting (17) in matrix form as

$$\frac{d\breve{x}}{dt} = A\breve{x} \tag{18}$$

where

$$\breve{x} = \left(\begin{pmatrix} x \\ y \end{pmatrix} \right) \text{ and } A = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}$$

Let $\breve{x} = \widehat{v}\,e^{\lambda t}$ be the trial solution of (17), where $\widehat{v}\,(\neq 0)$ is some fixed vector that needs to be determined.
Then

$$\frac{d\breve{x}}{dt} = \widehat{v}\,e^{\lambda t} = A\,\widehat{v}\,e^{\lambda t} \tag{19}$$

Canceling the non-zero scalar factor from both sides of (19), we now obtain

$$A\widehat{v} = \lambda\,\widehat{v} \tag{20}$$

From linear algebra, it can be easily concluded that λ is the eigenvalue of the matrix A, whose eigenvector is $\widehat{v}$, which is obtained by solving (21)

$$\det(A - \lambda I) = 0 \tag{21}$$

$$\rightarrow \begin{vmatrix} f_x - \lambda & f_y \\ g_x & g_y - \lambda \end{vmatrix} = 0 \tag{22}$$

$$\lambda^2 - (f_x + g_y)\,\lambda + f_x g_y - f_y g_x = 0 \tag{23}$$

$$\lambda^2 - \text{trace } A\,\lambda + detA = 0 \tag{24}$$

Let λ_1 and λ_2 be the two eigenvalues of the matrix A.
The necessary and sufficient conditions that λ_1 and λ_2 will be negative (if real) or have negative real parts (if complex) is $trace\ A = f_x + g_y < 0$

$$Det\,(A) = f_x g_y - f_y g_x > 0 \tag{25}$$

By Routh-Hurwitz criteria [57] in (25), we obtain the DNRCM/DCCN cubic polynomial (26) as the linearized model.

$$\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0 \tag{26}$$

Where $a_o a_1, a_2, \dots \dots a_n$ are non-zero coefficients.

The DNRCM/DCCN QoS stability criteria are now given as: $a_1 > 0, a_2 > 0, a_3 > 0, a_1 a_2 - a_3 > 0$. Within complex domains, (26) can be solved using the combined power series Frobenius method [58]. This implies that with Routh-Hurwitz criteria, established that DNRCM/DCCN QoS stability state, marginally stability state, and its unstable states can be determined leading to availability (ψ).

Also, recall that in **Fig. 3**, optimal internal switches are connected to cluster nodes to offer agility via reconfigurable scripts. The idea is to allow for a quick response to unpredictable application workload requirements from CPS-IoT general users in **Fig. 4**. Interconnecting the network to complex NOC nodes while providing improved fault tolerance via its routing

scheme is very desirable based on (13). Considering the issues established in **Fig. 5**, this work then explored a new interconnect topology that arranges NOC nodes in clusters of similar structures as shown in **Fig. 3**. We consequently interconnected the NOC clusters following a layered pattern of node coordinates. The reason is to reduce redundant connections between NOC clusters, thereby maximizing CPS connectivity at scale.

4.6. DCN Use Scenario

Using the **Table 1** design boundaries, in this work, we further identified resilience conditions for CPS workloads in **Fig. 5** to **Fig. 6**. These are unconnected laboratories whose workloads can be provisioned from **Fig. 3**. The completed CPS DCN for streams datasets is shown in **Fig. 7.** This depicts incoming, and outgoing users' traffic. The Lab workload is characterized using temporal traffic dependence and service size. Therefore, the design plan followed the earlier reference in **Fig. 3** satisfying replication, smart-provisioning, workload automation monitoring, and auditing. Besides, to migrate the Labs in **Fig. 5** and **Fig. 6**; cloud governance, risks, and compliance metrics are considered. Full-scale service-based resilience in the multi-complex DNRCM/DCCN model is deployed across the entire network. This deployment applied full interconnection interface intelligence on NOC nodes and layer-3 switches. The stability aspect of the QoS metrics is later presented using simulation SDN trace-files in modified Riverbed Modeler software [59] in Section 4.

Fig. 5. Unconnected pneumatics and hydraulics Labs, MCE, FUTO, 2021 (Authors Survey).

Fig. 6. Unconnected Lathe and Milling Labs, MCE, FUTO, 2021 (Authors Survey)

Fig. 7. Reference NOC for MCE virtual laboratory, FUTO, 2021
(Authors Testbeds, Swift Networks, Nigeria, 2021).

5. Experimental Analysis and Results

5.1. Evaluation Methodology

Riverbed Modeler version 17.5 [56] with scalable enablers plugins such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN) was used as the base-simulation platform [59]-[64]. We conducted the QoS experimental measurements for stable DNRCM topology. The benefit of the SDN approach is that it uses OpenFlow to decouple traffic flows from the Datapath and control plane. Most legacy workload network simulation tools like MATLAB Simevent, Omnet ++, Ns3, etc., hardly accommodate highly granularized NFV and SDN system of systems components. Therefore, we explored the scalable OpenFlow switch and OpenFlow controller. Both support a flexible, dynamic, and super-scalable toolkit that drives the faster realization of containerization, edge computing, and NFV for CPS SDN-enabled domains. For lightweight computation, the Riverbed simulation on Ubuntu OS is integrated with CloudSimSDN-NFV in this study. This provides library supports for SDN-NFV as well as robust edge SDN-enabled Fog-Clouds constructed from CloudSimSDN for CCPS. In evaluating the proposed DNRCM, tuning parameters in [49], [51] are employed considering the preliminary findings from the production testbed (UNN DCN). In the case of the Mininet, the ping command is used to confirm reachability in the SDN. For the production testing, iperf and Ethereal-Wireshark are used to gather real-time data streams. REST API interface is used to observe the real-time behavior of the workload on the network architecture. Both the traffic workload distribution and load balancing are considered at the core level for the QoS metrics evaluations.

In our validation, recursive_DCell and BCube Lab contexts are configured using a scenario configuration panel. **Fig. 8** shows the SDN virtualized topology for integrating the various CPS sites (*i...jn+1*) The implementation servers are Linux container models. The OpenFlow SDN controller is represented as an aggregation of CPS multipath link routes activating the forwarding engines to provision their processes/services. Each CPS site is a mesh pool of ethernet-based ports linking the SDN OpenFlow switch-controller. Mininet and Floodlight controllers were adopted for the implementation of the SDN controller with an embedded rule base deployed by the Open vSwitch engines. In Section 5.2, the #C++ SDN OpenFlow event trace file is used to construct the three instance scenarios for result analysis**.** For each of the scenarios, their attributes are equally configured before the execution. The reliability hazard function is introduced via an application configuration palette for OpenFlow multilayer

switches. The pod cluster server considered is based on Intel Xeon quad-Core 2.26 Hz and 48 GHz RAM. The various results obtained are discussed.
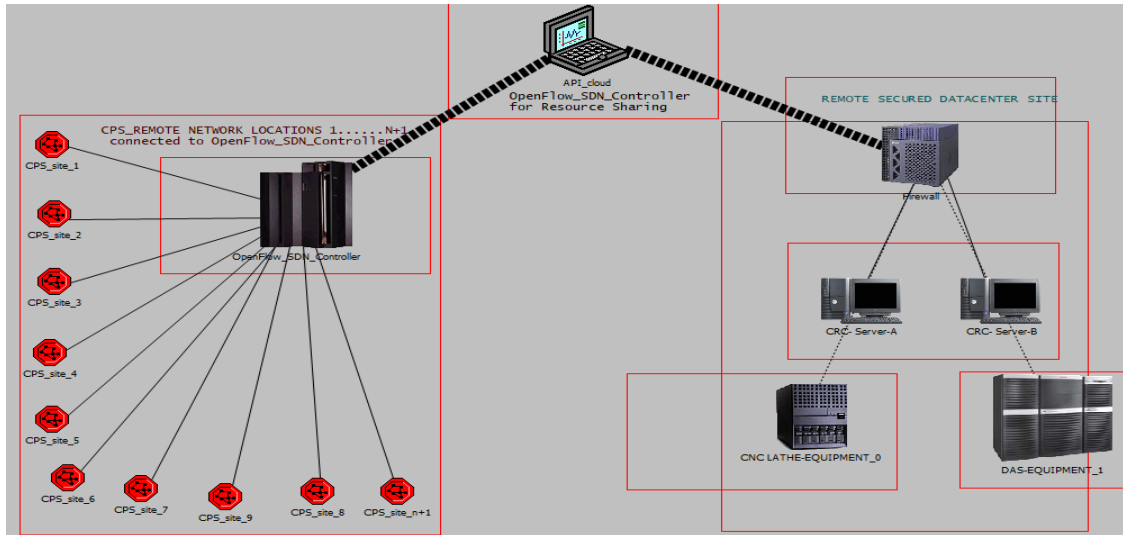

Fig. 8. DNRCM/DCCN SDN topology.

5.2. Results

The experiments focused on DNRCM logical isolation, and reliability hazard function for QoS provisioning. Pod cluster reliability analytics is shown to meet the QoS requirements. In this section, the selected metrics investigated include average delays/latency, throughput, fault-tolerance, utilization, availability, and scalability (i.e., dimensionless metrics). The results between the recursive models and the non-recursive schemes are discussed. **Fig. 9** to **Fig. 15** shows the results obtained from the DNRCM experiments. When compared to other topologies, the proposed DNRCM significantly increased network capacity because it has an optimal value of CPS-node bisection bandwidth. **Fig. 9a** shows the non-recursive queuing delays on the DNRCM. By introducing logical isolation, a lower queuing delay was observed in the DNRCM (19.65%) compared with the case of zero logical isolation (80.34%). **Fig. 9b** shows the DNRCM resource utilization under logical isolation compared to the case under no logical isolation. With logical isolation via the control plane on the system resources, 40% optimal utilization was observed. In the absence of isolation, 60% utilization is observed. The implication is that in a dynamic CPS network, the flow-rules placement will no longer overburden the SDN controller during the forwarding process leading to a lower utilization profile.
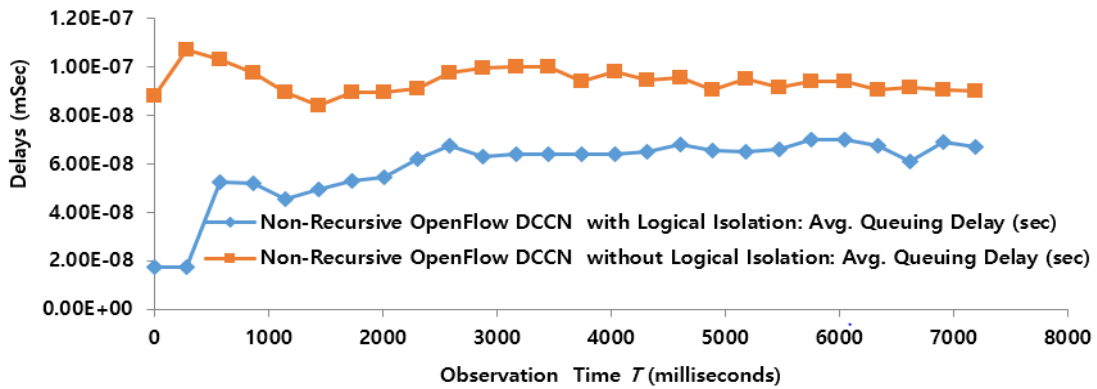
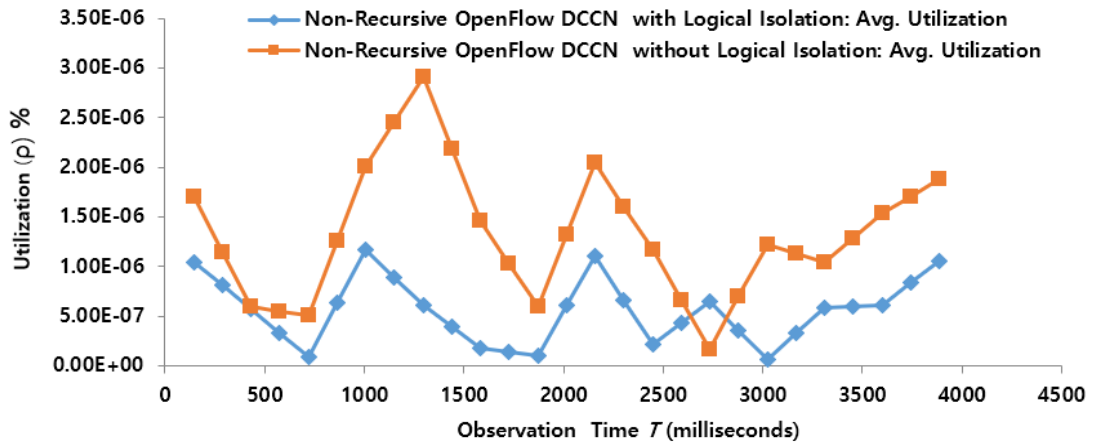**Fig. 9a.** DNRCM Delays for flow-rules placement requests.

Fig. 9b. DNRCM utilization.

In **Fig. 9a**, without isolation, the traffic density in **Fig. 3** will be extremely high leading to network saturation at the pod clusters. **Fig. 10** demonstrates the result of fault tolerance when logical isolation is introduced under single scenario port switch/node-degree setups. When one or more of the DNRCM system's components fail, fault tolerance deals with the ability to sustain the operation. By not logically isolating resources, fault tolerance gave 33.55%. With logical isolation in place, this accounts for 66.44%. The implication is that network tear and wear are remarkably high in the former while with self-stabilization, the system perseveres. When compared to a naively designed DCN, where even a minor failure can cause a total breakdown, the decrease in operating quality is proportional to the severity of the failure. In a high-availability scenario such as DNRCM, fault tolerance is overly critical for CPS applications needing graceful degradation. The reason is that the OpenFlow controller can proactively manage flow tables for all traffic traces and use redundant paths in various pod clusters.
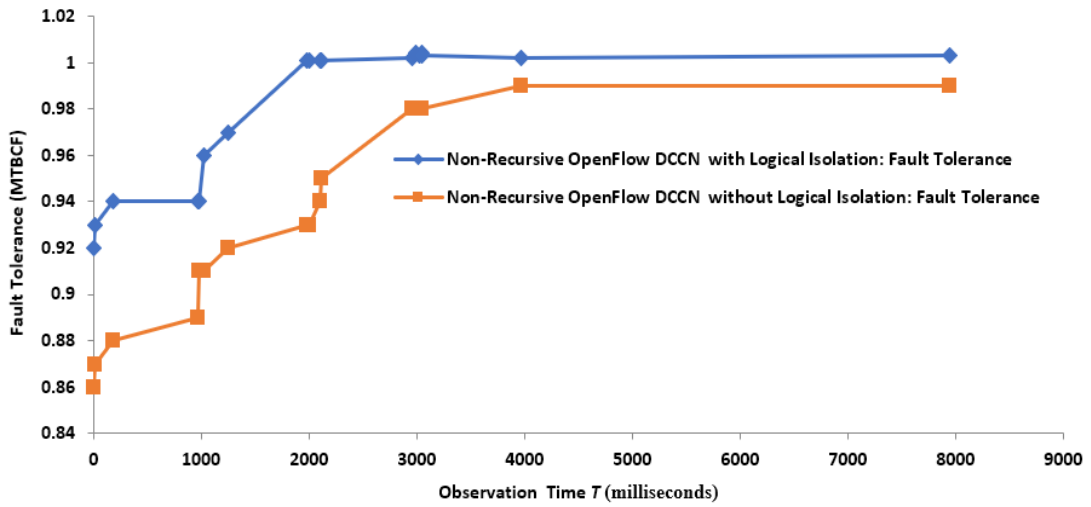
Fig. 10. DNRCM Fault Tolerance response.

Fig. 11 depicts an intriguing throughput observation when the reliability hazard function is considered. The response demonstrates that DNRCM support all traffic patterns following the mathematical models derived [3], [52]. As shown in **Fig. 11**, DNRCM maintained a reliable throughput of 38.30%, while the recursive BCube and DCell had 36.37% and 25.53% respectively. This implies that the aggregate bottleneck throughput (ABT) is the sum of all bottleneck flows in the DC. DNRCM has a good ABT. The ABT of a K-node DNRCM under All-to-All traffic patterns, for example, is high for a two-way communication link. ABT can reach optimal percent of total network capacity in DNRCM.
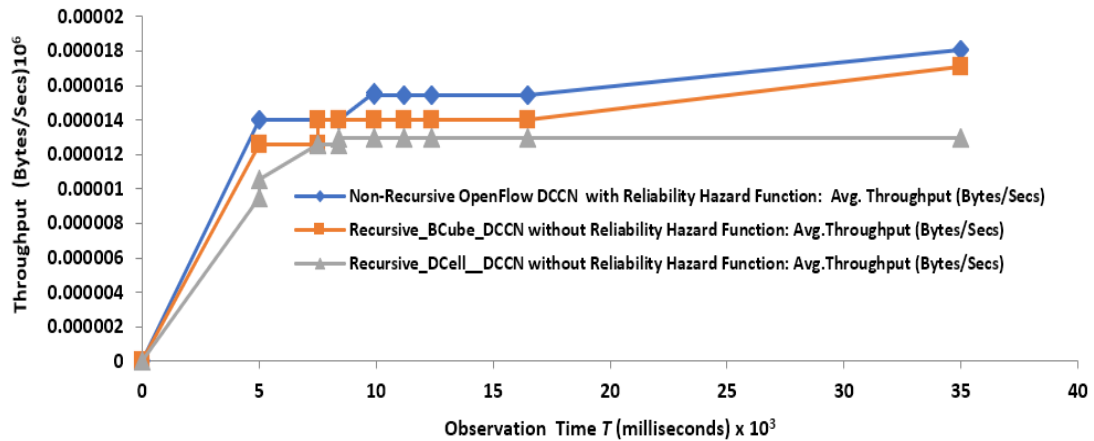

Fig. 11. DNRCM Throughput conditions under different configurations.

Fig. 12 shows the DNRCM scalability response under different scenario configurations. As depicted in the plot, by using a resilient OpenFlow multilayer switch, the scalability profile increases much faster compared to the use-case scenario of either recursive_BCube or DCell. The DNRCM topological configuration with smart redundancy, NFV, and hazard function improves incremental resource scalability by 40.00%, as shown in scalability comparison plots. BCube and Recursive DCell had 33.33 percent and 26.67 percent, respectively. Traffic workloads can be reliably processed with Tier-4 reliability hardware function (0.999999). As a result, only multilayer OpenFlow switches are recommended for scalable deployment.
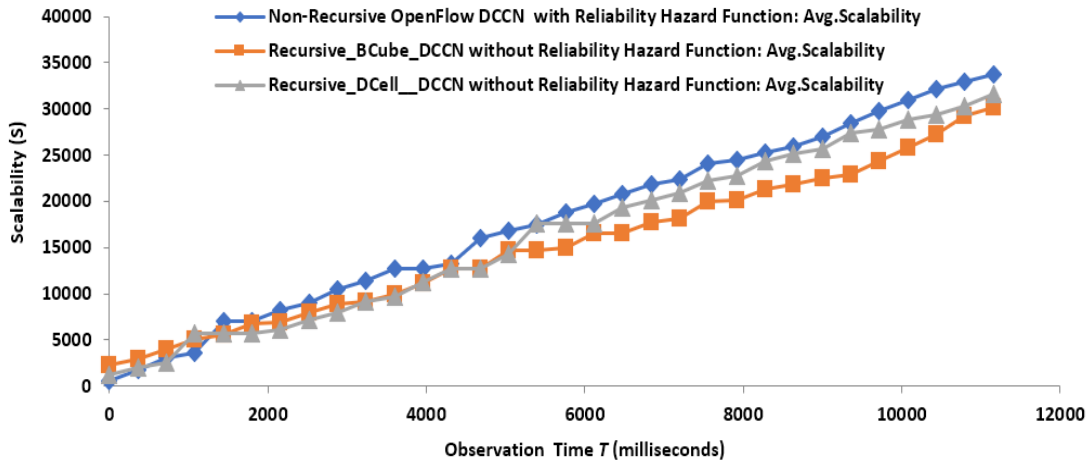
Fig. 12. DNRCM Scalability under different configurations.

Fig. 13 depicts the DNRCM availability under different configurations. In terms of availability, the DNRCM offers better uptime needed for Tier 4 and 5 services, (52.10%) compared with recursive BCube and DCell yielding 34.72% and 13.18% respectively. The row-based routing algorithm offers incremental construction and optimizes all the network metrics for near-zero downtime. **Fig. 14** shows DNRCM delay comparison under various scenarios of QoS provisioning. It was observed that the average delays for DNRCM, recursive BCube, and DCell are 32.81%, 33.44%, and 33.75% respectively. This implies that DNRCM offers shorter routes for a maximum number of hops. **Fig. 15** shows the reliability hazard function validation on resource utilization. According to the preliminary findings, DNRCM topology offers satisfactory attributes gives resolves DCell's double exponential scalability and BCube's high cost. This is resolved in the utilization concerns. When compared to existing topologies, DNRCM scales to massive sizes with 50.28% utilization without compromising performance even at an exceedingly high computational workload due to the logically isolated resources at scale. BCube and DCell yield 27.93% and 21.79% respectively under similar workload contexts.
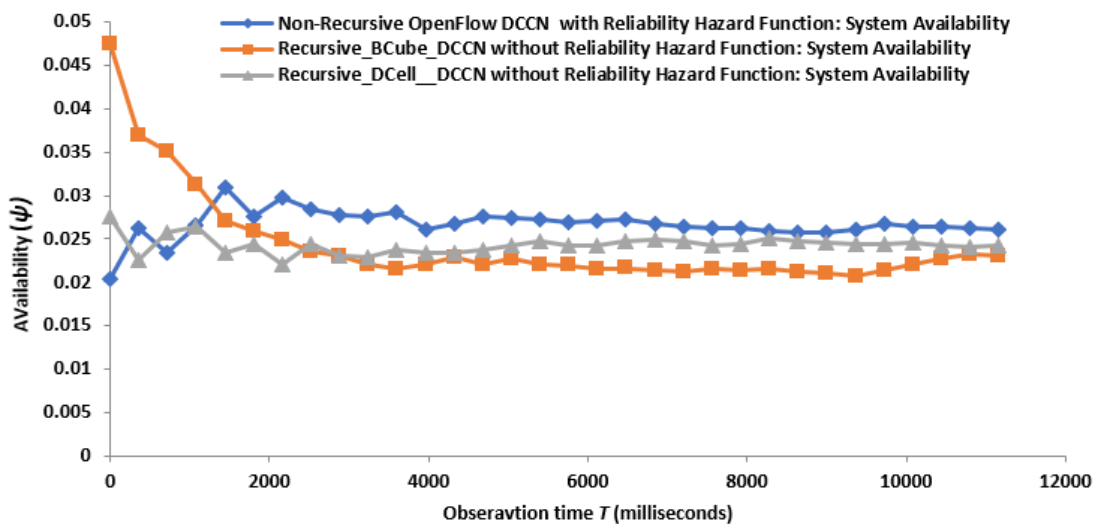

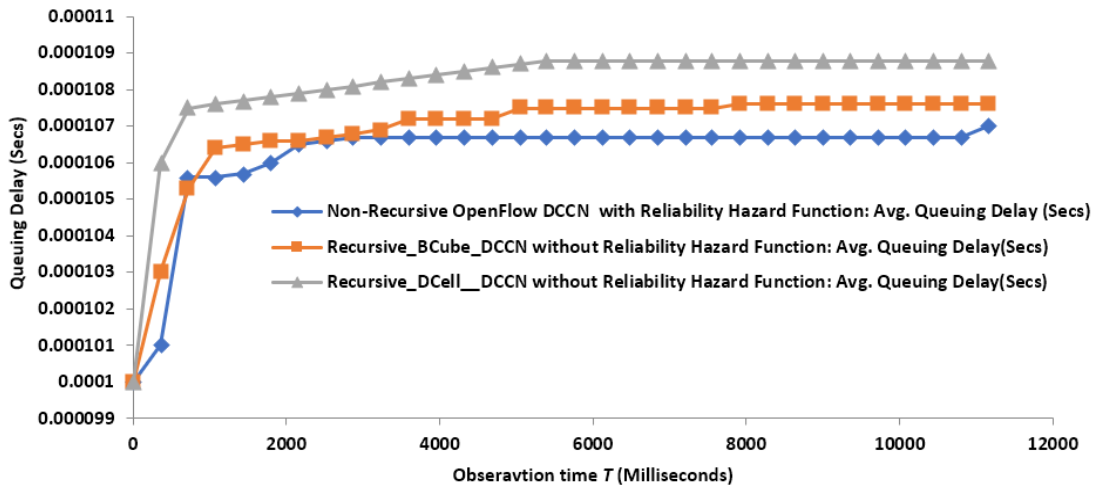**Fig. 13.** DNRCM Availability under different configurations.

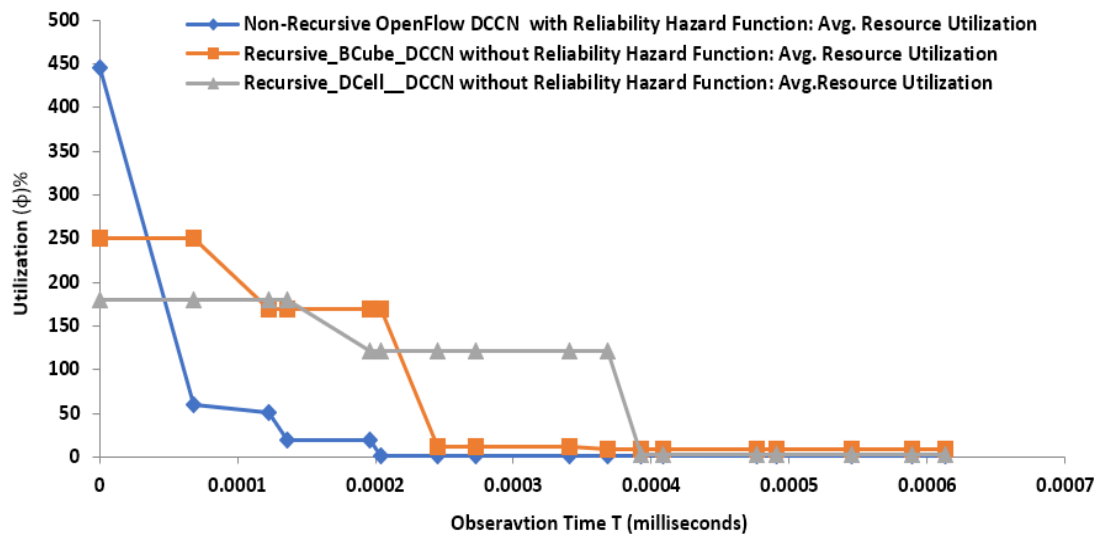**Fig. 14.** DNRCM Queuing Delay under different configurations.

Fig. 15. DNRCM Resource Utilization under different configurations.

So far, the metrics for logical isolation are factored in the SDN OpenFlow controller. Tables and II show the reliability control algorithm and the topological models. Both are responsible for broadcast failure control and QoS provisioning. OpenFlow switch was used for upstream and downstream workload bidirectional-forwarding-detection sessions. This happens with each neighboring Pod cluster in **Fig. 3 and Fig. 4**. Also, the metrics for the reliability hazard function, determine possible active or passive links leading to availability. The results show that the in-built series and parallel analytics in the neighboring internal and external switches offer higher fault tolerance (66.44%). In Table III, the controllers proactively deploy forwarding rules. The reason is to optimize workload traffic. By dynamically controlling the flow entries, the NRDCM uses the local and stability criterion to oversee traffic performance while balancing the QoS demands for system availability. **Tables 2 and 3** give insight into the effect of control automation and design topology on CPS applications.

Table 2. DCN Control Algorithm Results Summary

Reliability Control Scheme	Delays t (s)	Utilization η %	Fault Tolerance ξ (hours)
Non-Recursive OpenFlow DCCN with Logical Isolation (SDN)	19.65%	40%	66.44%
Non-Recursive OpenFlow DCCN without Logical Isolation (SDN)	80.34%	60%	33.55%

Table 3. DCN Topological Results Summary

NRDCN Architecture	Throughput (Bytes/Sec)	Scalability (β)	Availability (ψ)	Queuing Delay δ (sec)	Utilization (ρ%%)
Recursive BCube	36.37%	33.33%	34.72%	33.44%,	27.93%
Recursive DCell	25.53%	26.67%	13.18%	33.75%	21.79%
Proposed Non-Recursive DCN	38.30%	40.00%	52.10%	32.81%	50.28%

5.3. Limitations

DNRCM has few limitations in its current design for CPS. Massive traffic workload, especially the type that needs ultra-low latency may have penalty-cost because of reconfiguration issues in the OpenFlow SDN controller. Another issue is to scale DNRCM using a large DC space that will house scaled thousands of servers for computational analytics. This needs efforts in both fault injection models and best automation design practices. However, this is still part of an ongoing study. Finally, DNRCM has some concerns about its robustness since a single physical SDN controller offers a default single-point-of-failure challenge. A failure with an SDN controller will make a layer-3 switch to be constrained from routinely forwarding new data streams or packets leading to exponential failure. The research community can fix this problem by introducing chaos engineering after programming the layer-3 switch in form of an OpenFlow-hybrid controller.

6. Conclusion

This paper has proposed and evaluated a Cyber-physical DC topology called DNRCM. SDN Floodlight controller is implemented using smart load balancing controls between the CPS nodes and the destination sinks. System performance testing was done at the distribution core for dynamic load balancing on the network. Routh-Hurwitz stability criteria were used in the cubic polynomial formulation. DNRCM scales up DCN reasonably (especially with agile automation and QoS) by exclusively using smart links and robust multilayer switches. DNRCM provides high Scalability, aggregate throughput, good fault tolerance, short average queuing delays, and wide bisection availability. DNRCM optimizes the number of clusters connected directly with a distinct symmetric connection pattern. With the divided nodes into similar clusters and connections, an asymmetric coordinate system with minimal redundant interconnections between pod clusters was achieved. At a minimum, only Tier-4 DNRCM is recommended for service providers working towards resilient CCPS-DCNs. When compared to existing topologies, the proposed topology allows large-scale data centers to scale in size considering their computational activities without compromising performance. Future work will investigate lightweight provisioning and scheduling using container deployment and deep learning optimization schemes and the Fish growth model by Von Bertalanffy.

7. Future Research Directions

This paper has discussed the integration of physical laboratories using optimal control algorithms for a complex Cyber-physical network design. For smart access, a lower latency event-driven CCPS framework with support for CloudSimSDN-Network Function Virtualization (NFV) is presented. This facilitates the analysis and quick evaluation of resource provisioning specifically for Edge and Cloud Spine-Leaf layers. In terms of network process models, this work used the OpenFlow SDN controller to estimate the QoS metrics when juxtaposed with Mininet.11. Use-cases needing precision-based analytics for QoS parameters in the CCPS network infrastructure must be powered by Machine learning models. Deep learning and convolutional network security will be investigated considering efforts in [65]-[68] for GB/s optical DCNs in Smart grid ecosystems in future. Time-space complexity analysis will be developed and validated.

References

[1] D. Li, H. Zhao, M. Xu, and X. Fu, "Revisiting the Design of Mega Data Centers: Considering Heterogeneity Among Containers," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1503-1515, Oct. 2014. Article (CrossRef Link)

[2] X. Tao, K. Ota, M. Dong, W. Borjigin, H. Qi and K. Li, "Congestion-aware Traffic Allocation for Geo-distributed Data Centers," *IEEE Transactions on Cloud Computing*, pp.1-1, 2020. Article (CrossRef Link)

[3] K. C. Okafor "Development of a model for smart green energy management using distributed cloud computing network," Ph.D. dissertation, Dept. Elect. Eng., University of Nigeria Nsukka, 2017.

[4] W. Xing, D. Du, A. Bakhshi, K. -C. Chiu and H. Du, "Designing a Transferable Predictive Model for Online Learning Using a Bayesian Updating Approach," *IEEE Transactions on Learning Technologies*, vol. 14, no. 4, pp. 474-485, 1 Aug. 2021. Article (CrossRef Link)

[5] J. Son, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "SLA-Aware and Energy-Efficient Dynamic Overbooking in SDN-Based Cloud Data Centers," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 76-89, 1 April-June 2017. Article (CrossRef Link)

[6] M. Yuang, P. Tien, W. Ruan, T. Lin, S. Wen, P. Tseng, C. Lin, C.-N. Chen, C.-T.g Chen, Y. Luo, M. Tsai, and S. Zhong, "OPTUNS: Optical intra-data center network architecture and prototype testbed for a 5G edge cloud [Invited]," *Journal of Optical Communications and Networking*, vol. 12, no. 1, pp. A28-A37, 2020. Article (CrossRef Link)

[7] K. Zhang, C. Keliris, T. Parisini and M. M. Polycarpou, "Identification of sensor replay attacks and physical faults for Cyber-physical systems," *IEEE Control Systems Letters*, vol. 6, pp. 1178-1183, 2021. Article (CrossRef Link)

[8] N. Correll, N. Arechiga, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson, H. Liu, A. Patrikalakis, T. Robertson, J. Smith, D. Soltero, M. Tanner, L. White and D. Rus, "Building a distributed robot garden," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1509-1516, 2009. Article (CrossRef Link)

[9] J. Le, X. Lei, N. Mu, H. Zhang, K. Zeng and X. Liao, "Federated Continuous Learning with Broad Network Architecture," *IEEE Transactions on Cybernetics*, vol. 51, no. 8, pp. 3874-3888, 2021. Article (CrossRef Link)

[10] K. C. Okafor, "Dynamic reliability modelling of cyber-physical edge computing network," *Int'l J. of Comp and App, (IJCA)*, vol. 43, no. 7, pp. 612-622, 2021. Article (CrossRef Link)

[11] L. Guo, P. Congdon, "IEEE SA Industry Connections--IEEE 802 Nendica Report: Intelligent Lossless Data Center Networks," IEEE SA Industry Connections--IEEE 802 Nendica Report: Intelligent Lossless Data Center Networks, pp.1-44, 22 June 2021. Article (CrossRef Link)

[12] A. A. Obayi and K. C. Okafor, "Cloud-Fog Orchestration Infrastructure for Ante-natal Health Care Systems," in *Proc. of Int'l Conference in Mathematics, Computer Eng., and Computer Science (ICMCECS)*, pp. 1-6, 2020. Article (CrossRef Link)

[13] A. Al-Baiz, M. Abu-Amara, A. Mahmoud, M. H. Sqalli and F. Azzedin, "Internet access denial by higher-tier ISPS: A NAT-based solution," in *Proc. of 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 001004-001008, 2011. Article (CrossRef Link)

[14] C. C. Udeze, K. C. Okafor, C. C. Okezie, I. O. Okeke and C. G. Ezekwe, "Performance Analysis of R-DCN Architecture for Next Generation Web Application Integration," in *Proc. of IEEE 6th Int'l Conf. on Adaptive Science & Technology (ICAST)*, pp. 1-12, 2014. Article (CrossRef Link)

[15] D. Guo, J. Xie, X. Zhou, X. Zhu, W. Wei and X. Luo, "Exploiting Efficient and Scalable Shuffle Transfers in Future Data Center Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 997-1009, 1 April 2015. Article (CrossRef Link)

[16] X. Wang, A. Erickson, J. Fan and X. Jia, "Hamiltonian properties of DCell networks," *The Computer Journal*, vol. 58, no. 11, pp. 2944-2955, Nov. 2015. Article (CrossRef Link)

[17] S. Kan, J. Fan, B. Cheng and X. Wang, "The Communication Performance of BCDC Data Center Network," in *Proc. of 12th Int'l Conf., on Communication Software and Networks (ICCSN)*, pp. 51-57, 2020. Article (CrossRef Link).

[18] Z. Chkirbene, R. Hadjidj, S. Foufou and R. Hamila, "LaScaDa: A novel scalable topology for data center network," *IEEE/ACM Trans on Net,* vol. 28, no. 5, pp. 2051-2064, 2020. Article (CrossRef Link)

[19] W. Chen, B. Liu, I. Paik, Z. Li and Z. Zheng, "QoS-aware data placement for MapReduce applications in geo-distributed data centers," *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 120-136, Feb. 2021. Article (CrossRef Link)

[20] X. Guo, X. Xue, F. Yan, B. Pan, G. Exarchakos and N. Calabretta, "DACON: a reconfigurable application-centric optical network for disaggregated data center infrastructures [Invited]," *Journal of Optical Communications and Networking*, vol. 14, no. 1, pp. A69-A80, 2022. Article (CrossRef Link)

[21] H. Dong, A. Munir, H. Tout and Y. Ganjali, "Next-Generation Data Center Network Enabled by Machine Learning: Review, Challenges, and Opportunities," *IEEE Access*, vol. 9, pp. 136459-136475, 2021. Article (CrossRef Link)

[22] Z. Jia, Y. Sun, Q. Liu, S. Dai and C. Liu, "cRetor: An SDN-Based Routing Scheme for Data Centers With Regular Topologies," *IEEE Access,* vol. 8, pp. 116866-116880, 2020. Article (CrossRef Link)

[23] Y. Hong, Q. Tang, X. Gao, B. Yao, G. Chen and S. Tang, "Efficient R-Tree Based Indexing Scheme for Server-Centric Cloud Storage System," *IEEE Trans on Knowledge and Data Eng*, vol. 28, no. 6, pp. 1503-1517, 2016. Article (CrossRef Link)

[24] Z. Zhang, Y. Deng, G. Min, J. Xie, L. T. Yang and Y. Zhou, "HSDC: A Highly Scalable Data Center Network Architecture for Greater Incremental Scalability," *IEEE Trans on Parallel and Distributed Sys.*, vol. 30, no. 5, pp. 1105-1119, 1 2019. Article (CrossRef Link)

[25] Xx O. H. Karam, "Pruning Generalized Hypercube Interconnection Networks for Diameter Preservation: RedCube," in *Proc. of Int'l Conf on Comp., and Applications (ICCA)*, pp. 1-368, 2018. Article (CrossRef Link)

[26] N. G. Kini, M. S. Kumar and H. S. Mruthyunjaya, "A Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture," in *Proc. of IEEE Int'l Advan. Comp. Conf*, pp. 858-861, 2009. Article (CrossRef Link)

[27] Y. Huang, L. Lin and S. -Y. Hsieh, "A Fast $f(r,k+1)/k$-Diagnosis for Interconnection Networks Under MM* Model," *IEEE Transaction on Parallel and Distributed Systems*, vol. 33, no. 7, pp. 1593-1604, 1 July 2022. Article (CrossRef Link)

[28] Y. Zhang, J. Guo and B. Xu, "Optical Interconnection Network Based on Distributed Switching for Data Center Application," in *Proc. of Int'l Conf., on UK-China Emerging Tech (UCET)*, pp. 96-99, 2021. Article (CrossRef Link)

[29] T. Baptista, L. B. Silva and C. Costa, "Highly scalable medical imaging repository based on Kubernetes," in *Proc. of IEEE Int'l Conf., on Bioinformatics and Biomedicine (BIBM)*, pp. 3193-3200, 2021. Article (CrossRef Link)

[30] X. Xue et al., "ROTOS: A Reconfigurable and Cost-Effective Architecture for High-Performance Optical Data Center Networks," *Journal of Lightwave Technology*, vol. 38, no. 13, pp. 3485-3494, 2020. Article (CrossRef Link)

[31] Y. Liu, H. Gu and N. Wang, "HPSTOS: High-Performance and Scalable Traffic Optimization Strategy for Mixed Flows in Data Center Networks," *IEEE Transaction., on Cloud Comp*, pp.1-1, 2021. Article (CrossRef Link)

[32] W. -Y. Chiu, W. -K. Hsieh, C. -M. Chen and Y. -C. Chuang, "Multi-objective Demand Response for Internet Data Centers," *IEEE Trans. on Emerging Topics in Comp. Intell.*, vol. 6, no. 2, pp. 365-376, 2022. Article (CrossRef Link)

[33] S. Garg, K. Kaur, G. Kaddoum and S. Guo, "SDN-NFV-Aided Edge-Cloud Interplay for 5G-Envisioned Energy Internet Ecosystem," *IEEE Network*, vol. 35, no. 1, pp. 356-364, 2021. Article (CrossRef Link)

[34] X. -Y. Li, W. Lin, X. Liu, C. -K. Lin, K. -J. Pai and J. -M. Chang, "Completely Independent Spanning Trees on BCCC Data Center Networks With an Application to Fault-Tolerant Routing," *IEEE Trans. on Parallel and Distributed Sys.*, vol. 33, no. 8, pp. 1939-1952, 1 2022. Article (CrossRef Link)

[35] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang; Yunfeng Shi, Chen Tian, Yongguang Zhang, Songwu Lu, "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," *ACM SIGCOMM Computer Comm. Review*, vol. 39, no. 4, pp. 63–74, 2009. Article (CrossRef Link)

[36] D. Guo, "Aggregating Uncertain Incast Transfers in BCube-Like Data Centers," *IEEE Transactions on Parallel and Distributed Systems*, 28(4), 934-946, 1 April 2017. Article (CrossRef Link)

[37] Haitao Wu, Guohan Lu, Dan Li, Chuanxiong Guo, Yongguang Zhang, "MDCube: A High-Performance Network Structure for Modular Data Center Interconnection," in *Proc. of CoNEXT'09*, pp. 25-36, 2009. Article (CrossRef Link)

[38] M. Hamdi, "Keynote speaker: Massive data centers for future Cloud computing applications," in *Proc. of Int'l Conf. on Computing, Management and Telecoms (ComManTel)*, pp.1-2, 2013. Article (CrossRef Link)

[39] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang and S. Lu, "FiConn: Using Backup Port for Server Interconnection in Data Centers," in *Proc. of IEEE INFOCOM* 2009, pp. 2276-2285, 2009. Article (CrossRef Link)

[40] E. Baccour, S. Foufou, R. Hamila and A. Erbad, "Green data center networks: a holistic survey and design guidelines," in *Proc. of 15th Int'l Wireless Comm., & Mobile Computing Conf. (IWCMC)*, pp. 1108-1114, 2019. Article (CrossRef Link)

[41] I. A. Stewart, "Improved Routing in the Data Centre Networks HCN and BCN," in *Proc. of 2014 Second Int'l Symp., on Comp. and Networking*, pp. 212-218, 2014. Article (CrossRef Link)

[42] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks," in *Proc. of NSDI'10: Proc. of the 7th USENIX Conf. on Networked systems design and implementation*, pp. 19, 2010. Article (CrossRef Link)

[43] A. Singla, C. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking Data Centers Randomly," in *Proc. of 9th USENIX Symp. on Networked Sys Design and Implementation*, 2012. Article (CrossRef Link)

[44] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen and Y. Chen, "OSA: An Optical Switching Architecture for Data Center Networks With Unprecedented Flexibility," *IEEE/ACM Trans on Networking*, vol. 22, no. 2, pp. 498-511, 2014. Article (CrossRef Link)

[45] L. Huang, Q. Jia, X. Wang, S. Yang and B. Li, "PCube: Improving Power Efficiency in Data Center Networks," in *Proc. of IEEE 4th Int'l Conf on Cloud Computing*, pp. 65-72, 2011. Article (CrossRef Link)

[46] A. R. Curtis, T. Carpenter, M. Elsheikh, A. López-Ortiz and S. Keshav, "REWIRE: An optimization-based framework for unstructured data center network design," in *Proc. of IEEE INFOCOM*, pp. 1116-1124, 2012. Article (CrossRef Link)

[47] K. Zhu, Z. Zhang and F. Huang, "iCautz: A high-capacity and fault-tolerant intercontainer network," in *Proc. of 3rd Int'l Conf. on Comp. Science and Network Tech*, pp. 732-736, 2013. Article (CrossRef Link)

[48] T. Kellermann, F. Cañellas, R. González and D. Camps-Mur, "vL2-WIM: Flexible virtual layer 2 connectivity services in distributed 5G MANO domains," in *Proc. of Joint European Conf. on Networks and Comm & 6G Summit (EuCNC/6G Summit)*, pp. 413-418, 2021. Article (CrossRef Link)

[49] C. C. Udeze, K. C. Okafor, C. C. Okezie, I. O. Okeke and C. G. Ezekwe, "Performance Analysis of R-DCN Architecture for Next Generation Web Application Integration," in *Proc. of IEEE 6th Int'l Conf. on Adaptive Science & Tech (ICAST)*, pp. 1-12, 2014. Article (CrossRef Link)

[50] X. Li, C. -H. Lung and S. Majumdar, "Energy aware green spine switch management for Spine-Leaf datacenter networks," in *Proc. of IEEE Int'l Conf. on Comm (ICC)*, pp. 116-121, 2015. Article (CrossRef Link)

[51] K. C. Okafor, I.E Achumba, G.A Chukwudebe, G.C Ononiwu, "Leveraging Fog Computing For Scalable IoT Datacenter Using Spine-Leaf Network Topology," *J. of Electrical and Comp. Eng*, vol. 2017, pp.1-11 Egypt, 2017, Article ID 2363240. Article (CrossRef Link)

[52] A. Liu, Y. Sun and Y. Ji, "FSCOI: A High Fan-Out, Scalable, and Cluster-Based Optical Interconnect for Data Center Networks," *IEEE Comm. Letters*, vol. 23, no. 2, pp. 266-269, Feb. 2019. Article (CrossRef Link)

[53] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel and S. Sengupta, "Vl2: A scalable and flexible data center network," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 51–62, 2009. Article (CrossRef Link)

[54] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. of ACM SIGCOMM Conf. Data Commun. (SIGCOMM)*, pp. 63–74, 2008.

[55] NEC-Fault tolerant server, (Whitepaper). [Online]. Available: https://www.nec.com/en/global/prod/express/collateral/whitepaper/ft_WhitePaper_E.pdf (Retrieved 1/22/2022)

[56] Gyu-min Lee, Cheol-woong Lee and Byeong-hee Roh, "Riverbed modeler reinforcement learning M&S framework supported by supervised learning," in *Proc. of IEEE 2021 Int'l Conf., on Inform Networking (ICOIN)*, pp. 824-827, 2021. Article (CrossRef Link)

[57] M. A. Choghadi and H. A. Talebi, "The Routh-Hurwitz Stability Criterion, Revisited: The Case of Multiple Poles on Imaginary Axis," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1866-1869, July 2013. Article (CrossRef Link)

[58] J. Deutscher and N. Gehring, "Output Feedback Control of Coupled Linear Parabolic ODE–PDE–ODE Systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4668-4683, Oct. 2021. Article (CrossRef Link)

[59] R. Bruschi et al., "An SDN/NFV Platform for Personal Cloud Services," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1143-1156, 2017. Article (CrossRef Link)

[60] I. Farris, T. Taleb, Y. Khettab and J. Song, "A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems," *IEEE Comm. Surveys & Tutorials*, vol. 21, no. 1, pp. 812-837, Firstquarter 2019. Article (CrossRef Link)

[61] G. Faraci and G. Schembra, "An Analytical Model to Design and Manage a Green SDN/NFV CPE Node," *IEEE Transaction on Network and Service Management*, vol. 12, no. 3, pp. 435-450, Sept. 2015. Article (CrossRef Link)

[62] Z. Lv and W. Xiu, "Interaction of Edge-Cloud Computing Based on SDN and NFV for Next Generation IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706-5712, July 2020. Article (CrossRef Link)

[63] W. Zhuang, Q. Ye, F. Lyu, N. Cheng and J. Ren, "SDN/NFV-Empowered Future IoV with Enhanced Communication, Computing, and Caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274-291, 2020. Article (CrossRef Link)

[64] R. Flores Moyano, D. Fernández, N. Merayo, C. M. Lentisco and A. Cárdenas, "NFV and SDN-Based Differentiated Traffic Treatment for Residential Networks," *IEEE Access*, vol. 8, pp. 34038-34055, 2020. Article (CrossRef Link)

[65] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li and J. Rao, "Dynamic Flow Migration for Embedded Services in SDN/NFV-Enabled 5G Core Networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2394-2408, 2020. Article (CrossRef Link)

[66] A. M. Zarca, J. B. Bernabe, A. Skarmeta and J. M. Alcaraz Calero, "Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1262-1277, 2020. Article (CrossRef Link)

[67] R. Mu and X. Zeng, "A Review of Deep Learning Research," *KSII Transactions on Internet and Information Systems*, vol. 3, no. 4. pp. 1738-1764, 2019. Article (CrossRef Link)

[68] W. Xiong, X. Jia, D. Yang, M. Ai, L. Li, and S. Wang, "DP-LinkNet: A convolutional network for historical document image binarization," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 5, pp. 1778-1797, 2021. Article (CrossRef Link)

Kennedy Chinedu Okafor is a Senior Member of IEEE, USA. Kennedy holds Ph. D in Digital Electronics Engineering (Cloud Computing/Green Energy Specialization) from the University of Nigeria, Nsukka in 2017. Kennedy previously worked with National Agency for Science and Engineering Infrastructure, (NASENI-ELDI-FMST), Nigeria as a Senior R&D Engineer (2006-2014) before joining as a teaching researcher at the department of Mechatronics Engineering, Federal University of Technology, Owerri, Nigeria. Kennedy has various Certifications including CCNA, CCNP (Data Center, R&S), CompTIA A+, Server+, DevNet, and IBM Cloud Certifications. Kennedy has published in reputable/top impact factor venues including IEEE, Wiley, Elsevier, Springer, IGI-Global, and Taylor & Francis, among others. Kennedy has served on over 50 Technical committees (IEEE workshops/Conferences, Tetfund, National Assembly, & various consortiums). He has won several grants, reviewed over 280 papers, managed over 285 editorial tasks, (both in Scopus & Web of Science), and has presented papers at both Int'l and local Conferences. Kennedy is the Chair, of IEEE Consultants Network, Nigeria. He is a visiting senior research Fellow at FEDCODTTEN, Nigeria. He is a Senior Research Associate with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. Kennedy is an Adjunct facilitator with the World Bank Africa Center of Excellence for Sustainable Power and Energy Development (ACE-SPED), University of Nigeria, Nsukka, Nigeria. He collaborates with Cisco Systems NetAcad, IBM Global University Program, Africa, and Autonomous & Intelligent System Laboratory (AIS-Labs), School of Mechatronic Systems Engineering, Simon Fraser University, BC-Canada. Kennedy is a Fellow of the African Scientific Institute, USA. His research interests cover Advanced Cloud Data Center designs, FPGA/IoTs, AI, Green Energy Analytics, and 4IR domains.

Omowunmi Mary Longe is presently a Senior Lecturer and Chair of the Smart Power and Energy Research Group in the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. She received her Doctor of Engineering (D.Eng.) in 2017 from the University of Johannesburg, South Africa in Electrical and Electronic Engineering Science. Her M.Eng. and B.Eng. degrees were also obtained in Electrical and Electronics Engineering in 2011 and 2001 respectively from the Federal University of Technology, Akure, Ondo State, Nigeria. She is also the Vice Chair of IEEE PES Women in Power executive committee. She has published more than forty papers at referred journals and conferences. She is also a reviewer for ISI-listed journals and referred professional Conferences. She has served as member of Technical Programme Committees for local and international conferences. She is a senior member of the Institute of Electrical and Electronics Engineers, (IEEE), and South African Institute of Electrical Engineers (SAIEE). She is also a member of Nigeria Society of Engineers (NSE), Association of Professional Women Engineers of Nigeria (APWEN), Society of Women Engineers (SWE) U.S.A., etc. Dr. O. M. Longe is also registered with the Engineering Council of South Africa (ECSA). Her research interests include Smart secured systems, Renewable Energy Technologies, Microgrid design, e-Mobility, Energy Poverty, Distributed Energy Storage, Demand Side Management, Demand Response, Smart Energy Management, Energy Efficiency, Distributed Energy Generation, Rural Electrification, Energy and Gender Poverty Nexus, Energy and Power System Analysis and Dynamics, Women in Engineering, Engineering Education, Smart City Energy and Transportation, 4IR and Energy Systems, and Food, Energy and Water (FEW) Nexus.