IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2017, accepted January 26, 2018, date of publication January 31, 2018, date of current version March 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2800408

iPRDR: Intelligent Power Reduction Decision Routing Protocol for Big Traffic Flood in Hybrid-SDN Architecture

AMMAR K. AL MHDAWI AND HAMED S. AL-RAWESHIDY, (Senior Member, IEEE)
Wireless Networks and Communications Centre, Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences,
Brunel University London, London UB8 3PH, U.K.

Corresponding author: Ammar K. Al Mhdawi (eepgaka1@brunel.ac.uk)

ABSTRACT Analysing data centers energy consumption is the main step toward building a reliable infrastructure. Evidently, data centers consume a large number of billions of gigabytes information to the point that putting tremendous pressure on energy suppliers. Every internet activity involves a huge amount of data that need to be stored in a cloud data center somewhere, not forgetting the Internet of Things applications and other social media services that produce an extraordinarily large scale of big data that require high processing and analysis. Moreover, current data centers consume about 3% of global electricity supply, which is about 416.2 TWh of power that world data centers consumed in 2016. In this paper, we have developed an intelligent power reduction decision routing protocol (iPRDR) in a medium scale hybrid-software-defined network data center environment. The proposed iPRDR protocol approach is to dynamically segregate big traffic and route it to a high index processing devices with a power-optimal selected path. The protocol approach is to decrease the overall power consumption of the whole network, as well as to reduce the failure rate in each device that may occur due to a high level of link congestion and elevated temperature. The experimental results show that uplink utilization has been reduced by 8.33% and power consumption levels reduced by 0.85kw/day, which is equivalent to 34.9% of the operational power. In Addition, the high-raised temperatures have been dropped from high range 50+ C° (high critical) to mid 40+ C° (high warning). The effectiveness of the proposed approach was verified experimentally using a virtualized testbed platform.

INDEX TERMS Power reduction decision routing protocol (iPRDR), performance index (PI), layer metric, uplink utilization, power consumption model, power management, OF switches.

I. INTRODUCTION

In data centre environment, the power consumption of network devices plays an exceptional part in the communication and stability of network connectivity. The supplied traffic correlate with the energy consumption of the resources and their capability [1]. Looking at a traditional topology of network design, we find that there are many challenges that tag along with it, many of them are the increased costs of devices' yearly maintenance along with being considered a highly complex environment which requires highly skilled engineers to work with in conjunction with over-provision which means that devices require continuous hardware upgrades for future demands, and hence these demands are costly and lead to increase of power utilization per device, the software- defined network (SDN) was introduced to tackle these issues. SDN is a new networking model design that proposes the segregation of control plane and data plane, to such a degree allowing operators to have a centralized point of management that can administer the routing policies in its topology table and unicast forwarding rules to the open flow switches (OFS) which handle forwarding mechanism. The communication between the SDN controller and the OFS is via a protocol called open flow protocol, which manages the messages and queries between the controller and the OFS. The communication channel between the controller and OFS is secured via a Secure Socket Layer (SSL) connection. Presently, the immediate shift to a full SDN architecture is impossible because the full deployment requires many modifications and changes in the network infrastructure. Moreover, the replacement of the current vendor devices is very costly not mentioning the service contracts which have to be met with the vendors. In the meanwhile, the introduction of full SDN has to be incremental due to technical and financial reasons. The middle solution for that is to construct a hybrid SDN environment that combines the features of traditional and SDN topology.

Simultaneously, power consumption in the telecommunication and service provider networks are increasing expeditiously due to the vast big data traffic that is being generated from a variety of platforms and service such as IoT and social media platforms [2]. Thus, in this study, we examine the power consumption and fail probability of traditional network devices due to high overload traffic generated from such platforms. Our proposed protocol algorithm works along with Dijkstra algorithm (shortest path algorithm). Dijkstra is used in the open shortest path (OSPF) protocol which is considered the most commonly used intra-domain routing protocol which we will be deploying in our proposed architecture scheme. Dijkstra algorithm is considered as an iterative algorithm, meaning that it operates in iterations. At the first iteration, the algorithm search for the nearest neighbor interface from the source device which must be directly connected to it. At the second iteration, the algorithm search for the second-closest device from the source device. The device must be a neighbor of the nearest second found device. The process continues at Nth tries of iterations until the path is fully up and complete. The entire path calculations are based on the cost of the path. It also gives the predecessor device along a least-cost path from the source node. Load balancing can be achieved via OSPF as it is considered a standard functionality of the Cisco IOS. When a router learns multiple paths to a specific destination via OSPF, it installs the route with the lowest cost in the routing table, and since there are many routes with the same cost to the destination, the router will immediately start to load balance the traffic.

The critical big data traffic that requires high and immediate processing will be congested at traditional devices due to the extreme load that is exhausting the CPU core of the traditional router and bottleneck the uplinks [3]. In our experimental testing, we used SDN controller with failover capability. The SDN controller is deployed with traditional architecture in form of a hybrid design where all devices are connected together via multiple layers in a systems called layered approach. The layered approach is the basic foundation of the traditional data centre design which consists of the core layer, aggregation layer, and access layer. The core layer provides packet switching backplane for traffic going in and out of the data centre. The core devices run interior routing protocols such as OSPF and load balances traffic to the next domain layer. The aggregation layer provides service module integration and internal routing. Finally, the access layer which is connected to the servers and storage devices [4]. In the proposed mentioned design, a new power consumption model will be implemented which is based on the domain layer calculations. Furthermore, the proposed protocol will segregate critical traffic on traditional devices which will lead to a reduction of congestion levels on device uplinks and devices core processing units and increase the Quality of Service (QoS) levels of big data traffic which will be handled via a high index performance switches for sophisticated data processing. The main contributions of our study can be summed up as follows:

- We propose a new path calculation algorithm named Power Reduction Decision Routing Protocol (iPRDR). The iPRDR consist of domain layer calculation for each layer of devices in the data centre. The algorithm aggregate and segregate big traffic to be routed towards the high indexed devices with the optimal-power path based on a calculated metric value. The result of this technique will reduce power on each traditional device as that will help cut down power bill cost based on the dropped wattage value along with a reduction in traditional device CPU core temperature and drop in congestion levels which can lead to a reduced probability of device failure.
- We model the power consumption of the SDN controller and OFS devices using power optimization technique. We extend the modeling technique presented in [15] to work with big data traffic and SDN functionality so that the power consumption and link utilization congestions levels are calculated based on a given set of parameters. The extension of our power calculation formulas is based on calculating the regular traffic versus big load traffic and their correlation with the actual power consumed per device. In our calculations, we focused on the components that are power-hungry which may cause future system failures. The used set of mathematical parameters are listed in the notions table with their corresponding description.
- We conducted extensive testing on real operational data centre testbed using a virtual machine instances which consisted of traditional routers and open flow switches and SDN controllers. We observed that a significant amount of congestion levels on uplinks were dropped which prompt to power consumption reduction per device. Furthermore, the failure rate of each device in times of big traffic has been reduced to avoid any outage disaster that could happen in a production network.

The reminder of the manuscript is arranged as follows. In Section II, previous work is summarily outlined. In section III, SDN data centre architecture is described. In Section IV, testbed design and algorithm formulation are presented and defined. In Section V, the power consumption model calculations are illustrated. In Section VI, the experimental results and performance analysis are presented. Lastly, the conclusion and future work in Section VII.

II. PREVIOUS WORK

Numerous studied conducted by many researchers from different communities regarding power saving in a network environment in the prospect of green networking which is environmental friendly approach and has economic advantages as well. The energy consumption approach can be implemented in wired or wireless networks and both are similar to each other in regards to the concept of implementation.

Cianfrani *et al.* [5] presented a novel network-level strategy that is based on the modification of the link-state OSPF routing protocol. The concept is applied by shutting down the power on the low level utilized links. The author proposed a solution in a three-stage algorithm which is compatible with traditional link-state routing protocols such as OSPF. The algorithm allows the implementation of power reduction via using the topology link data that are exchanged via routers in the OSPF protocol. The energy-aware routing (EAR) algorithm is based on Dijkstra algorithm concept which proposes that only subset of routers shortest path tree are used to select routing paths. The EAR algorithm showed that 50% of the links can be powered off thus reducing power consumption.

Correia *et al.* [6] provided a holistic approach for energy efficient radio networks. The author added that in a radio network at a component level, the power amplifier component is complemented by transceiver supporting advanced energy management which is the main key to efficient radio communications. The author presented a framework for energy efficiency evaluation analysis by identifying the levels of network component that need to be addressed to save power.

Mumey *et al.* [7] illustrated the power down approach for idle links and nodes including routers and switches. The paper discusses routing data in a given set end-to-end communication network where the total energy consumption is reduced by shutting down any link that is not being used in the bundled links. The author presented two polynomial-time heuristic algorithm which is routing with power down problem (RPP) shortest path and RPP-Tree.

In [8], Azari and Miao conducted an energy consumption modeling approach with better battery lifetime analysis and transmit power control for massive MTC over radio networks. The paper provided uplink scheduling solutions for MTC over single-carrier frequency division multiple access (SC-FDMA) systems. Also exploring the MTC scheduling that is based on Max-Min lifetime fairness analysis. The obtained results showed that the modeling power consumption of MTC can greatly prolong the network lifetime.

In [9], Zhou *et al.* address power control and spectrum resource allocation issue in SWIPT-based D2D networks. The author proposed two algorithms, first is a preference establishment algorithm based on Dinkelbach method and Language dual decomposition and second is energy efficient algorithm based on Gale-Shapley algorithm. The concluded work was joint power formulation to optimize EE performance of D2D pairs and the energy harvested by CUEs concurrently.

Some research papers investigated the problem of power consumption reduction in software-defined networks. In [10], Giroire *et al.* presented an approach for energy-aware routing (EAR) in SDN networks. The EAR process put the unused links into sleep mode to save power. The SDN controller can identify traffic matrix and calculate the best route to satisfy QoS with the minimum energy consumption for backbone devices while respecting capacity limitations on links. The model is formulated using Integer Linear Program (ILP) by using energy efficient heuristic algorithm. The author provided that by using this approach, the power consumption percentage was almost close to the one used with classical EAR algorithm.

In [11], Prete *et al.* proposed GreenMST, a simple open flow controller that generates a loop-free L2 architecture according to specific metric by defining a spanning tree protocol in the open flow network that helps with power saving. GreenMST works by turning off the inactive interfaces which will end up preventing loops that cause lots of energy consumption. The target of this study is large-scale data centers where the loop-free topology is considered a very important and essential part of the network for less power consumption.

In [12], Levin and Canini applied the benefits of implementing the Panopticon, an architecture that combines the traditional network with SDN controller. The paper focused on the main point that network service providers and operators have to deploy SDN incrementally in order to build more confidence with its credibility and operation to handle a large production network with minimum outages. Moreover, the budget constraints that come with deploying a full SDN can be to a great extent costly. The mentioned Panopticon architecture abstracts the network into a logical SDN and hides the traditional devices and map them to the logical SDN abstraction for the underlying hardware.

In [13], Xie *et al.* proposed E^{3}MC, a mechanism to improve the data centre network energy consumption via elastic multi-controller SDN. The power optimization for the data and control plane are considered by utilizing routing and dynamic control mapping and consolidating the traffic load onto a small set of devices and shut down the redundant ones to save power. The author concluded that the E^{3}MC server heavy computation load is a limitation for scalability. Hence, it can be extended to work in a distributed environment within each SDN controller.

In [14], Vissicchio *et al.* illustrated an experimental work that proposes a generic sequence-computation approach based on two algorithms that calculates an operational sequence that guarantees a proven consistency throughout the update in pure SDN and hybrid-SDN structure as well. The author was able to prove that the proposed procedure calculates a safe operational sequences that update the maximal number of devices without overload. The update overhead was reduced by 80-100%.

In [15], Ohsugi *et al.* presented a model for energy reduction by Named Data Networking (NDN) in a multicore software routers for optimizing the prefix matching and caching which considered a power-hungry process. The author proposed a power consumption module for the core CPU using optimization techniques. The model was verified using NDN Forwarding Daemon source code. The author added that the power consumption of a router's CPU is proportional the traffic load and thus this power proportionality is very important to reduce the power consumption rate.

The above-mentioned research studies on traditional network, SDN and Hybrid-SDN architecture intend to reduce the power consumption levels of network devices and to minimize the congestion on uplinks and core processing units. The drop in congestion levels lead to better QoS and minimizing the levels of fail rate and packet drops for each device in the data centre. Our study proposes a new approach for big data traffic management in a hybrid-SDN architecture. The outcome of the implemented protocol is to reduce power consumption in the entire data centre and to minimize the failure rate to avoid any future outages in the production network.

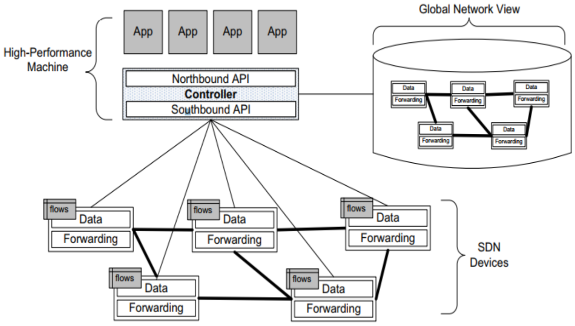


FIGURE 1. SDN architecture overview.

III. SDN DATA CENTRE ARCHITECTURE

A. SDN ANATOMY AND OPERATION

Software-Defined Networks (SDN) is a new emerging technology that is manageable, cost-effective and can suite high-data applications in today's modern data centre. In [16], SDN definition per Open Networking Foundation (ONF) is the physical decoupling of the control plane from the forwarding plane where a control plane connects to all forwarding planes, thus having a global view of the network topology. The separation of the planes enables the control plane to become programmable via a dynamically written code that can automate the entire network from a centralized point of management. Moreover, SDN can simplify network topology and process of operation due to its ability to send instructions to all devices in the network instead of using multiple vendor-specific equipment and protocols. The protocol that governs the communication between the SDN controller and OFS is called Open Flow protocol [17]. Open Flow protocol has the ability to update, add and delete flows from the OFS flow table. The flow table in the switch consists of flow entries of matching fields, counters and set of instructions that will be matched with the packet when it arrives at the OFS. If a matching entry is identified, the instruction set will be executed. Seemingly, SDN provides special application programmable interfaces (API) [18], [19] on the controller to permit the automation of the controller. These interfaces are called North Bound Interface (NBI) and South Bound Interface (SBI). The interfaces are used to distinguish between whether the interface is connected to the application layer on the NBI or to the open flow switches on the SBI. As we see in Fig. 1 The SBI is the Open Flow interface that the SDN controller use to automate and program the network OFSs. While the NBI API allowing applications such as Java or Python to be connected to the controller which can be easily injected into the controller for quick and easy network management. On the right side of the SDN controller, we can see that a maintained global topology is defined. This topology helps the controller to calculate the best path in an engineered automated manner. The injected code can implement modifications in the network when the changes are required.

When OFS device receives a packet, it overlooks its flow table to find matching criteria. The flow tables have been installed previously by the controller into the OFS at the initial stage. When a match statement found, the OFS will take appropriate action by forwarding the packet to the appropriate interface. If there is no match found, then the OFS will send a query to the controller for further consultation on what to do with the packet. Continuously, in reference to our overview of the open flow operation, the SDN controller maintain a total overview of the entire topology and generate routing policy decisions and perform load balancing on the open flow switches. To examine the internal structure of the controller, controllers often come with their own internal application platforms such as internal firewall, routing engine, and load balancer. These applications are in form of code that is merged with the controller. Fig. 2, shows the anatomy of the control plane. The figure presents the main application modules that are considered the core functions of the controller.

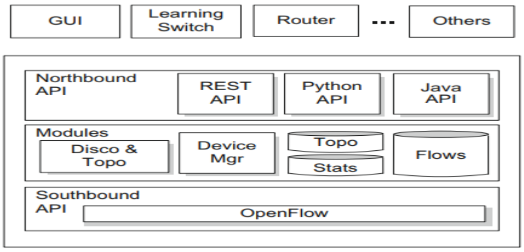


FIGURE 2. Internal structure of the control plane.

The core functionality of the control plane can be illustrated as follows:

- **Discovery and Topology Manager**: perform the discovery of the end-user device such as laptops, PC, printers, as well as network changes tracking with device management.
- **Device Manager**: implement the discovery of new network devices that comprise the network infrastructure such as switches, routers, access points.
- **Topology Manager**: sustain the total information about the current interconnections of the network and statistics.

- **Flow Manager**: maintain a link database of the flows and synchronize them with the OF switches. Additionally, it maintains a per-flow statistics about the OF switches to track device status.
- **REST API**: is a set of instructions written in any code that can perform specific functions on the SDN controller to request, modify and get information via HTTP protocol.

B. HYBRID-SDN IMPLEMENTATION CONCEPT

The Hybrid-SDN design as we mentioned earlier is considered to be an incremental step towards a full SDN architecture in the future. The Hybrid-SDN model is an approach to make traditional topology operate with SDN network in the same production environment. The OF switches are controlled via SDN controller, however, the traditional switches run a link-state routing protocol internally such as OSPF protocol for routing packets. The hybrid zone of operation for the traditional devices that run OSPF protocol start by sending hello packets [19] to the connected neighbors to determine if there is a node present on the link or not. If a neighbor is connected, then the neighbor enabled OSPF will respond to the hello packet with a hello-reply packet to establish adjacency. Ones the adjacency is established, the routers start to exchange link state advertisements (LSA) between each other. The LSA contains the cost and subnet of each directly connected links. Any received LSAs will be flooded to the other connected routers except the interface that the LSA came from. Ones the LSAs are received, the routers will start to build their topology table which is called Link State Data Base (LSDB) table. The database will hold information about the topology. In large-scale networks, The LSBD contain the only partial overview of the topology. Furthermore, the OSPF use Dijkstra algorithm to find the shortest path (SP) tree to the destination. From the SP tree, the shortest best path will be installed in the routing table. The main route will have some backup routes as well to take over when the main path fails. Using Dijkstra-OSPF protocol for large-scale networks could be very sophisticated and would add high overhead on the router. The solution to that was to partition the entire OSPF topology into areas to minimize the calculation times of SP algorithm. Moreover, the gradual steps towards a full SDN data centre topology are impossible at the moment due to many risks and economical constrains at the present time. Thus, the merging of SDN with traditional architecture is the best current solution. The OF switches are controlled by the SDN controller which manage flow table updates and operation, despite the fact that the traditional networks will be running OSPF for our scenario. In Fig. 3 we present an example of a scenario case of the SDN interconnected components with traditional devices. The solid lines represent Ethernet links that carry regular traffic operated by OSPF protocol and the orange color line represents the SDN links where open flow protocol operates and forwarding rules get installed on the OF switches via the controller. Many scenarios of interconnection designs can be implemented depending on the needs of the data centre.

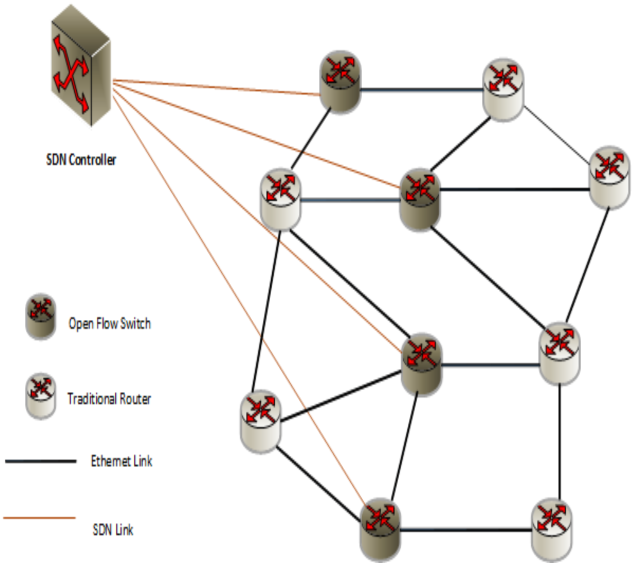


FIGURE 3. SDN and traditional topology interconnect.

IV. TESTBED DESIGN AND ALGORITHM FORMULATION

A. PROPOSED HYBRID-SDN ARCHITECTURE

We implemented the spine-and-leaf topology as the traditional topology for our experimental work. The spine-and-leaf topology [20] is a common data centre architecture in these days. Spine-and –leaf design is considered a two-tier architecture where the spine switches connect to the Top of the Rack Switches (ToR), which eventually connects to the servers in the rack. The Spine layer is the backbone of the testbed and is in charge of connecting to all the leaf switches. With this architecture, no matter where the destination is server is connected, the traffic will always cross the same number of devices to get to the server. In the above- mentioned architecture, we added open flow switches and SDN controllers to simulate the hybrid-SDN design. As we see in Fig. 4 we build a medium scale data centre environment with 48 switches including traditional and open flow switches and 36 servers. The devices are in form of virtualized machines instances that run on a Linux server. The spine layer consists of 29 switches that are interconnected with each other in a full mesh network to provide 100% redundancy in case of link or device failure. The leaf layer consists of 18 ToR switches. In our experiment, the SDN that was used is the open daylight controller (ODL) [21], which provide routing policy directions to the OF switches. The tracking of link congestion and utilization levels was via setting up a remote cacti server that runs simple network management protocol (SNMP) to track the status and flows of each device.

We simulated the big data traffic using a network traffic generator with up to 100Gbps of big traffic supplied on the uplinks via one or two data centres (DC1&DC2). The remaining traffic was regular traffic with variance levels of packet rates going inbound on the main network. The supplied traffic was logged per each device via the cacti server and we simulated the maximum congestion levels on uplinks of the device until we reach the packet drops levels.

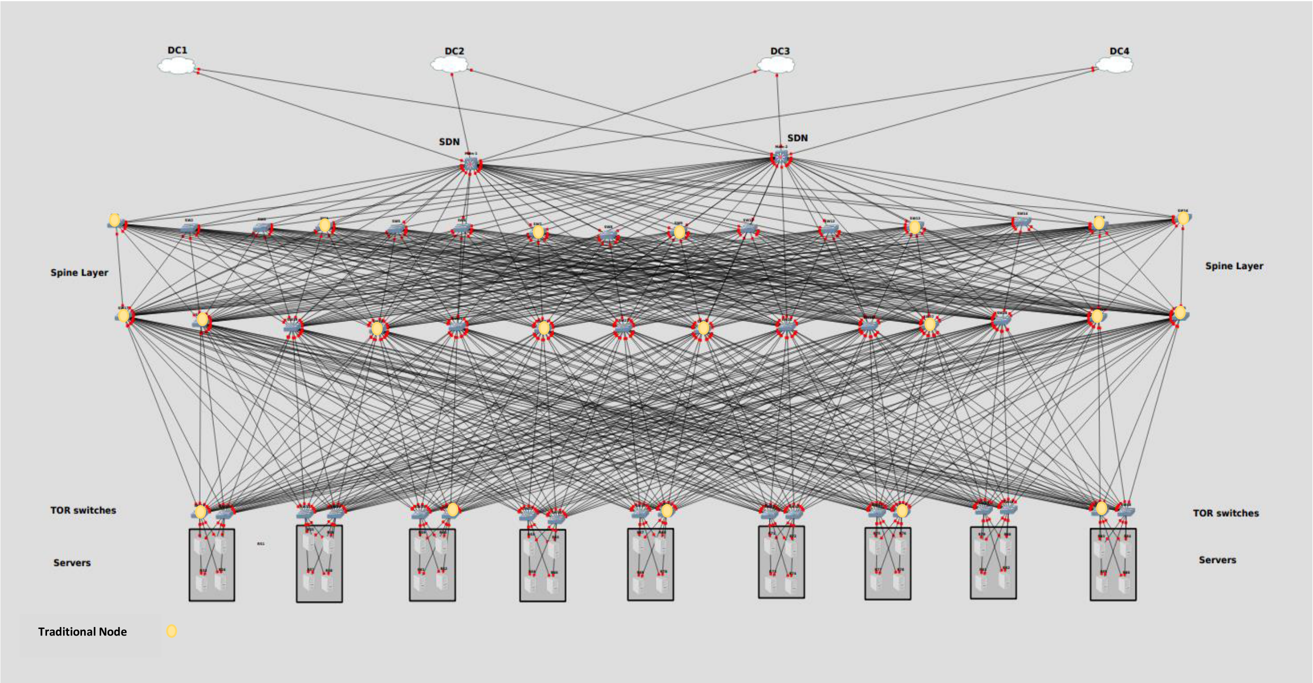


FIGURE 4. Testbed data centre design.

Link and device monitoring were implemented before and after adding the SDN controller to the network. We assumed that metric cost on all links is 100 and that OSPF will load balance on all downlink interfaces. We have noticed that after congesting the network with big traffic that many devices were prone to fails due to the high overhead on the uplinks and router's CPU. Many routers operational temperature has been raised significantly due to the high energy consumption per device which may put the devices at risk of failure.

B. ROUTING ALGORITHM FORMULATION

In our study, we propose a iPRDR algorithm that will optimize the uplink utilization levels on the hybrid network and reduce power consumption in each device. In Algorithm (1), our calculations and route management will be dependent on domain layer. We start by representing out our algorithm with $G = (D, L)$, where D represents the number of the device that we have in the network and L are the links that interconnect the devices. The input parameters to the algorithm are the highest capacity rate on the uplink UL_Cap_{high} and the lowest capacity value on the down link DL_Cap_{high}. We assume that there is a high-end level routing machine index in each domain layer called *d_performance index*, the performance index is a value that represents the device with the highest processing end to end capability. Each device in our proposed design has a performance index value that varies from its neighbor device. The device with the highest index value has the highest processing capability. As we all know that in any data centre there are different vendor technologies that vary from each other in term of capability, processing, resiliency, and performance. In our algorithm, we try to segregate the traffic from the devices with low index values and aggregate it to the nodes with the high index processing units. At the source node, the distance $d[r]$ is zero, and the distance to m node is unknown yet. Our algorithm operates in a form of layer per layer calculations. There are two main counters that we will use to perform domain layer Node calculations are known as i and j. The counter will go through any device in the network to retrieve the parameters for metric calculations. The metric values will be stored in a two-dimensional array called *metric[k, i]*. Each domain layer will calculate the metric and send it up the uplink to the controller for routing decisions.

However, the SDN controller must select a route based on the metric values. The controller will conduct a comparison for each metric value in the array. The highest value will be the best candidate to route the traffic. Furthermore, the highest metric value means that the next hop device has the highest capability of processing the big data. In accordance with the relationship between the device capability of handling such traffic and the traffic supplied, the initiation algorithm adjusts the performance index based on the uplink and downlink capacity. Furthermore, we propose that the OF switches will send the status of the traditional switches that run OSPF to the controller for routing table optimization. The OSPF table optimization starts if there was a big traffic to be flooded onto a traditional router, then the router will be prone to fail, thus the controller will be able to check the performance index of that switches and re-route traffic to the appropriate device for better energy reduction and to prevent any outages that could occur in the network. When a big traffic is detected on

Algorithm 1 Initiation and Metric Calculation

Input: $G = (D, L)$, UL_Cap_{high}, DL_Cap_{high}, *d_performance index*
Output: *metric[k.i]*

```
1.  $d[r] \leftarrow 0$; $d[m] \leftarrow \infty, \forall r \neq m, r, m \in D$
2.  Initialize $k \leftarrow 0$; /* initialize counter to zero*/
3.  Initialize $i \leftarrow 0$; /* initialize counter to zero*/
4.  input $UL\_Cap_{high}, \forall l \in$ uplink layer
5.  input $DL\_Cap_{high}, \forall l \in$ downlink layer
6.  input d_performance index
7.  for $\forall r \in D$ domain do
8.    calculate $metric[k.i] \leftarrow (UL\_Cap_{high} + DL\_Cap_{high})^*$ d_performance index
9.    Increment $i \leftarrow i + 1$
10.   if i <= 15; go to step 12/*considering that we have 15 nodes per domain*/
11.     else exchange all metric[k.i] values with upstream layer
12. Increment k = k + 1
13.   If k = 4, deliver packet to directly connected device
14.   Else; go to step 4
15. Endif
16. Endfor
```

Algorithm 2 Route Selection Process

Input: *metric[p]*, $\forall metric[p] \in$ *downstream layer*
Output: *Identify highest metric[p] for traffic forwarding*

```
17. Initialize $p \leftarrow 0$
18. If metric[p] > metric[p+1], then metric[p+1] = metric[p]
    /* comparison of metric values in the array*/
19. Increment p $\leftarrow$ p+1
20. If p > 15; forward traffic to the next node with highest metric node; then go step 4
21. Else; go to step 18
```

a traditional OSPF router, the router will use the traditional Dijkstra algorithm to calculate the shortest path tree. After the calculation is done. The routing metrics of traditional router are sent to the SDN controller as well. The SDN controller will work as an optimizer for the routing table by comparing its calculation with the traditional router calculations. If the path set by the traditional router $\phi_{entry}[j]$ has not satisfied the parameters of SDN routing policy β_{SDN}, then the SDN will overwrite the OSPF table with the β_{SDN} path as we see in Algorithm 3.

The SDN will work as optimizer on the traditional topology to avoid any low-weight path from carrying high-weight traffic that may impact the network. In Fig.5, the following flowchart will illustrate the process of operation for the three algorithms as follows:

Algorithm 3 Overwriting OSPF Routing Table

Input: $\alpha_{\mathrm{load}} \in$ big traffic, β_{SDN}, $\phi_{\mathrm{entry}}[\mathrm{j}]$, $\mathrm{UP}_{\mathrm{traditional}}$
Output: *modification on Table$_i$*

```
1. for i = 1 to $\alpha_{load}$ iterations do
2. while $\alpha_{load}$ is detected on $UL_{traditional}$ do
3. For j = 0 to K iteration do/* scanning routing entries*/
4. Search $Table_i$ for $\phi_{entry}[j]$
5. If $\phi_{entry}[j] == \beta_{SDN}$
6. Forward traffic $\rightarrow \phi_{entry}[j]$
7. Else Overwrite $\phi_{entry}[j] = \beta_{SDN}$
8. Forward traffic $\rightarrow \phi_{entry}[j]$
9. End if
10. End for
11. End while
```

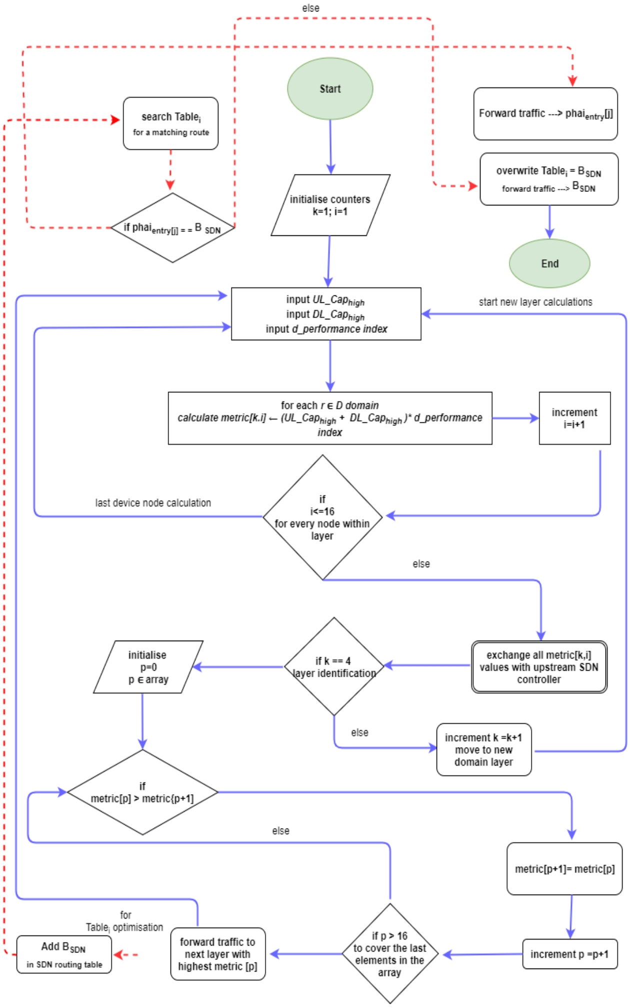


FIGURE 5. iPRDR algorithm process.

V. POWER CONSUMPTION FORMULATION

In our study, we developed a power consumption model of the energy consumption in our hybrid-SDN network. The formula that we modelled will focus on the power consumption in specific components such as CPU cores, fan as the power consumed on the NIC card that is in

TABLE 1. Notions used in the problem.

$G=(D, L)$	Network G, where D is the set of network devices and L is the number of physical links.
$d[r]$	The distance at the first node
α **load**	The big traffic supplied on the traditional device
β **SDN**	Optimal-power path to destination
ϕ **entry**$[j]$	Installed route in OSPF table
υ_{node}	Router/switch device
τ_{core}	The number of core in CPU unit
$temp_k$	The temperature readings of the core
$\phi_{l,core}$	The max power consumed by a critical core
$\delta_{l,fan}$	The max power consumed by a traditional fan
$\beta_{p,core}$	The max power consumed by a traditional core
$\Delta_{p,fan}$	The max power consumed by a critical fan
ψ_{utlz}	The real utilization level measured in the testbed
P_{op}	The max operational power consumed by the NIC card, fan and cpu.
N_{device}	The total number of devices in the topology
Λ_{cpu}	The total CPU units
$\Phi_{threshold}$	The max inbound traffic threshold
Ω_{fail}	The fail rate of the device

correlation with the packet flow per second. Respectfully, we will present the total power consumption formula that reflects the main energy drain components as that will be our focus in this paper. The model reflects the big data on the operating traditional router. The consumed power should be a function of the supplied traffic load. The load represents many parameters in the router such as CPU, Line Card (fan and NIC cards). Based on the great work that was conducted in [15], we extend and develop the formulation to correlate with our proposed algorithm and testbed. We define the general power that is being consumed by a router υ_{node} as follows:

$$P_{node}(b_{load}, \sigma_{nic}, \eta_{fan}) = p_{cpu}(b_{load}) + p_{nic}(\sigma_{nic}) + P_{fan}(\eta_{fan}) \quad (1)$$

The main four power consumption components in the traditional device are the b_{load} which represents the big data traffic bytes that are queuing in the memory units for further processing. The power consumed by each element in the device is defined as follows:

- $P_{cpu}(b_{load})$: is the power consumed by the CPU, measured in watt which correlates with the amount of large traffic supplied.
- $P_{nic}(\sigma_{nic})$: is the power consumed which measure the incoming gigabit packet rate.
- $P_{fan}(\eta_{fan})$: is the utilized power that measures the power consumed by the entire module fans. The speed of the fan correlates with the congestions levels of CPU and the ram.

In our proposed routing protocol we intended to detect and reduce big load traffic that will be supplied on the traditional devices. The segregation of big traffic will lead to less link utilization and lower the overhead on the cpu which eventually will have a significant impact on the power consumption levels. The total full power efficiency of the cores, can be expressed as follows:

$$P_{core\_total} = \sum_{k=1}^{N} \left(\frac{b_{load}}{\tau_{core*}\Lambda_{CPU}} + \tau_{core} * temp_k\right) \quad (2)$$

Table 1 represents the notions used in the algorithms formulation. As we notice in Eq (2), the reduction of the big load traffic on traditional devices will reduce the power consumed per CPU cores, thus, will lead to reducing the core temperatures which eventually will lead to low power consumption. We define the critical traffic as Crt_{load}, the critical traffic inbound can be expressed as follows:

$$Crt_{load} = b_{load} - \operatorname{Re} g_{load} \quad (3)$$

Where $\operatorname{Re} g_{load}$ is the regular traffic that is being processed by the traditional network for devices deployed with OSPF protocol. In this case, we can illustrate in Eq (4) that the b_{load} traffic can be expressed as the power consumed in the core per CPU unit as follows:

$$Crt_{load} = \tau_{core} * \Lambda_{cpu}\left(P_{core} - \sum_{k=1}^{N} \frac{temp_k}{\Lambda_{cpu}}\right) - \operatorname{Re} g_{load} \quad (4)$$

The total power of the testbed per line card can be modeled with the following equation including SDN and traditional devices. High-index devices are indicated here with a critical abbreviation to distinguish them from the traditional nodes as follows:

$$\begin{aligned} P_{total} = {} & \left(\sum_{l=1}^{Crt_{device}} \phi_{l,core} * K_{core} + \delta_{l,fan}\right) \\ & + \left(\sum_{p=1}^{R_{device}} \beta_{p,core} * U_{core} + \Delta_{p,fan}\right) \\ & + \sum_{j=1}^{R_{device}+Crt_{device}} Pakt_{rate} * \omega_{nic} \end{aligned} \quad (5)$$

Where ω_{nic} is the total power consumed by the NIC for processing 1 x ip packet. Additionally, $\Delta_{p,fan}$ is the power consumed by the core fan per one cycle. The total actual measured power in regards to the utilization levels can be expressed in Eq (6) as follows:

$$P_{actual} \approx \sum_{e=1}^{N_{device}} \begin{bmatrix} \left(P_{op_{cpu}} - P_{idle_{cpu}}\right) * \psi_{utlz} + P_{idle_{cpu}} \\ + \left(P_{op_{fan}} - P_{idle_{fan}}\right) * \psi_{utlz} + P_{idle_{fan}} \\ + \left(P_{op_{NIC}} - P_{idle_{NIC}}\right) * \psi_{utlz} + P_{idle_{NIC}} \end{bmatrix} \quad (6)$$

Where ψ_{utlz} is the utilization rate on the device. P_{op} is the operation maximum power in time of full congestion. By subtracting the operation power from the idle state, we can get the current operating power which then multiplied by the inbound utilization rate to represent how much actual power is being consumed. In the same manner, it also applies to other device components such as NIC card and fan. The actual power in Eq (6) is an approximate value because there are other components that we did not include for the reason that in our experiment we are focusing only on the major power consumption units. We can note that the actual power is in a proportional relationship with the incoming load. We can summarize the power consumption in two forms as in the following relationship:

$$P_{actual} \begin{cases} \rightarrow high\ \Omega_{fail} & if\ \alpha_{load} > \Phi_{threshold} \\ \neq P_{\max}, \rightarrow low\ \Omega_{fail} & u\sin g\ iPRDR \end{cases} \tag{7}$$

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

We implemented our iPRDR protocol in both cases experimentally using the virtualized testbed architecture described in Fig. 3. Moreover, we tested our algorithm using MATLAB simulation for calculating the power-optimal path to the destination. We started our testbed design by depicting the hybrid network architecture. The data collection was calculated per each device by using Secure Socket Shell (SSH) connection to each device and then login into the configuration mode of each console and collecting data while running our protocol. The data collected was focused on a variety of parameters such as, power consumption, temperature, utilization rate and the effect of our proposed protocol on these parameters. The following graphs will record the behavior of the above-mentioned parameters with both traditional topology and our proposed iPRDR protocol respectively.

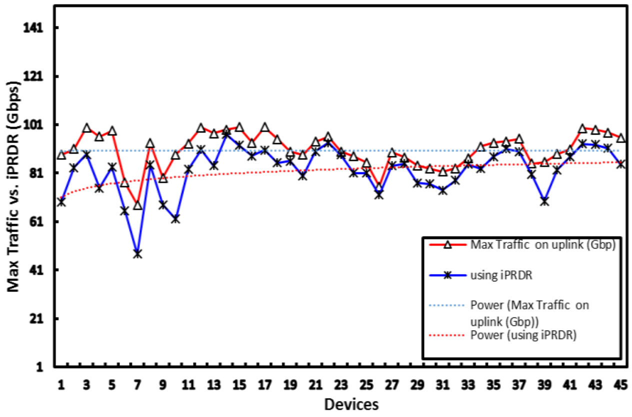


FIGURE 6. **Max traffic α_{load} vs. iPRDR protocol.**

In Fig. 6 the max traffic supplied α_{load} is 100Gbps. The random traffic flow was generated between 60Gps-100Gbps. The performance of our proposed iPRDR protocol as we see provide a status of stability and traffic drop on each device to sufficiently stabilize the congestion.

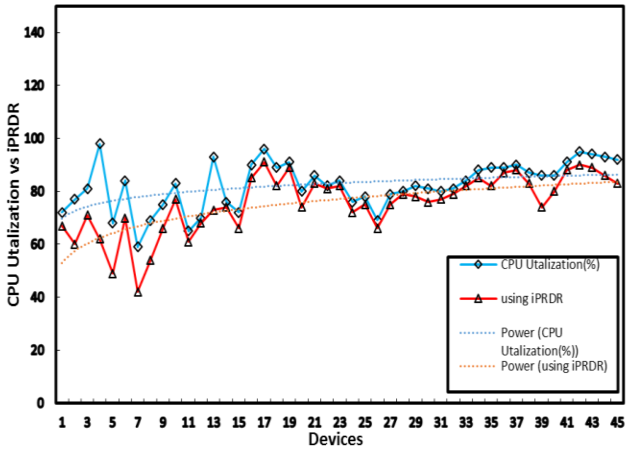


FIGURE 7. **CPU utilization ψ_{utlz} vs. IPRDR protocol.**

Consecutively, in Fig. 7 the impact of iPRDR algorithm is obvious on the CPU utilization ψ_{utlz} per each device. As α_{load} drop, the CPU utilization gradually decreases, thus less power will be consumed on the device. The ψ_{utlz} can be decreased more depending on the $\alpha_{\mathbf{load}}$ that is incoming per second.

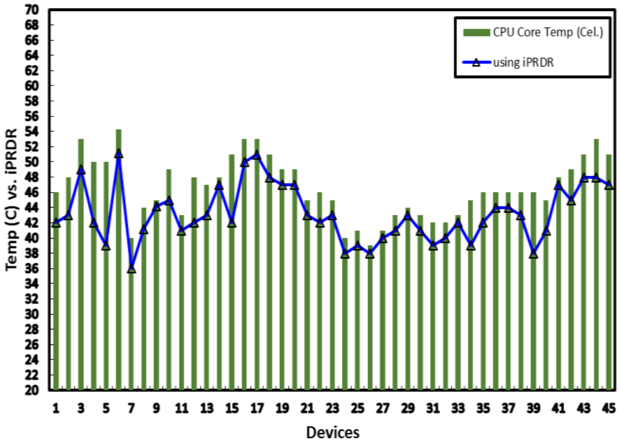


FIGURE 8. **Operation temperature vs. dropped percentage.**

In Fig. 8 the high inbound cause $temp_k$ of the core increase to a certain level that enters in the critical state where a device is prone to failure due to extreme overheating. Our proposed protocol iPRDR can reduce the temp to shift the temp from the high critical state to warning state which assists in preventing Ω_{fail} from increasing. In our experiment, we did not reach the 60C° as the amount of traffic supplied and traffic types need to be over 100Gbps which our test environment cannot provide due to limited resources. However, we were able to simulate the critical environment as we see in the Fig (8) above. In Table 2, the device temp warning caution are listed per vendor specification as below:

TABLE 2. Device temperature warning signals.

high warning	high critical	high shut
40C° - 50C°	50C° - 60C°	60C° - 75C°

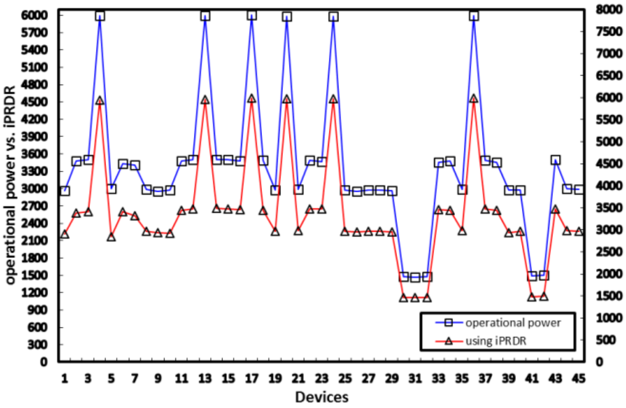


FIGURE 9. Operational power vs. power drop using iPRDR.

In Fig. 9, P_{actual} represents the maximum actual power consumed by the devices. Energy consumption is monotonously decreased when implementing iPRDR. The devices in the testbed vary in the operating power. Some of them operate in 6000 watts, the other operates in 3000 watts and 1500 watts depending on the device type. These devices are randomly located in the architecture as in any current data centre.

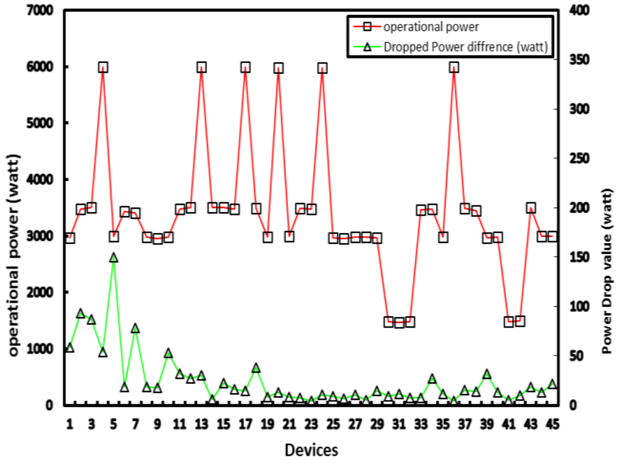


FIGURE 10. Operational power in regards to the dropped percentage at each device.

Moreover, in Fig. 10 and Fig. 11, the graphs illustrate the difference in wattage and traffic drop in regards to each device respectively. The difference values are very crucial for the device. Since these simple drops can affect the device healthy operational status and will decrease the probability of failure rate for better network operation. The distribution of fail points is described in Fig. 11 were we notice the high fails and their correspondent value when implementing our proposed protocol.

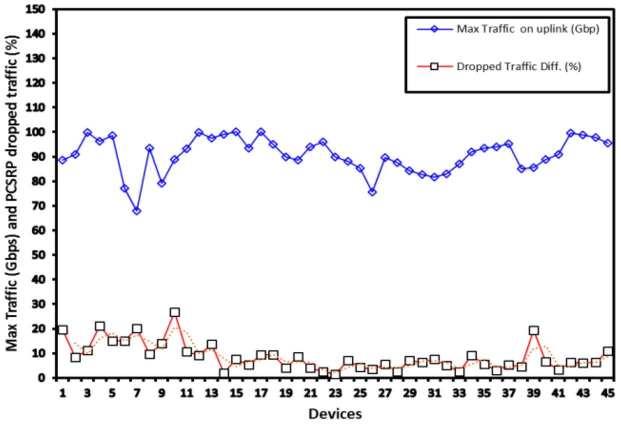


FIGURE 11. β_{load} traffic with the drop rates.

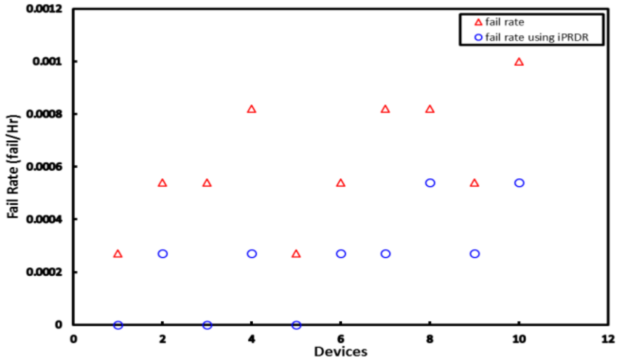


FIGURE 12. Fail rate Ω_{fail} distribution before and after implementing iPRDR.

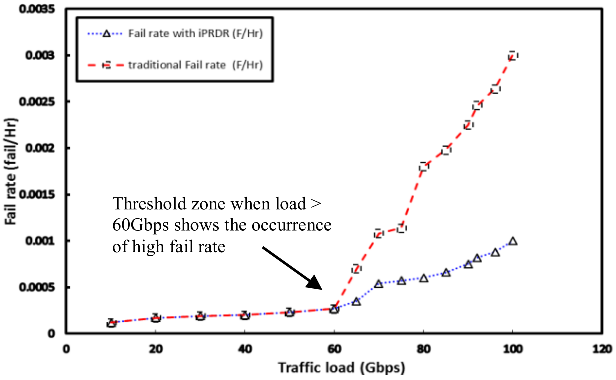


FIGURE 13. Fail rate Ω_{fail} curve before and after implementing iPRDR in accordance with the Max traffic (Gbps).

In Fig. 13 According to our results, we can notice that the threshold zone fluctuation starts when traffic is above 60Gbps. Furthermore, the increase in the fail rate is a fast rapid increment which is proportional to traffic inflation as well. However, in Fig 12, with using our proposed protocol, we can confirm the fail rate is slightly increasing which indicates a better network management process to keep the data centre operation healthy with less possible outages.

Moreover, in Fig 14, we can notice the reduction in traffic caused by iPRDR on the traditional devices close to 47Gbps drop rate.

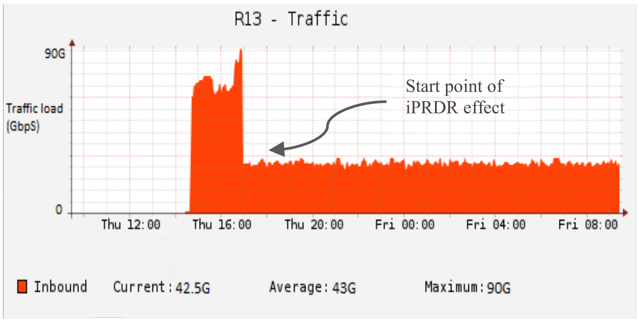


FIGURE 14. Sample test on a link shows the congestion drop to a stable level with implementing iPRDR.

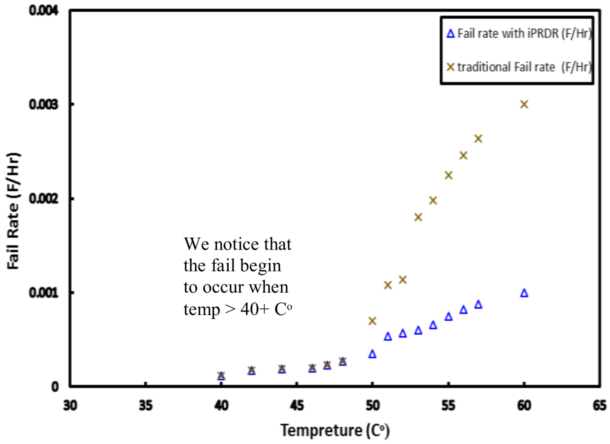
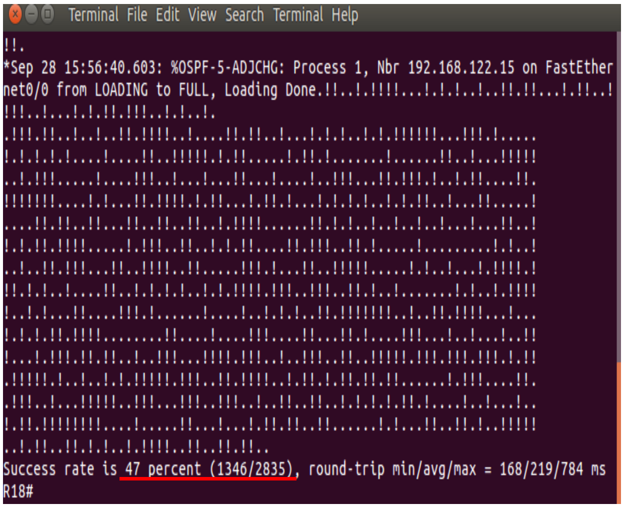


FIGURE 15. Approximate steady fail rate with iPRDR compared to the traditional environment.



```
!!.
*Sep 28 15:56:40.603: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.15 on FastEther
net0/0 from LOADING to FULL, Loading Done.!!..!.!!!!...!.!.!..!..!!.!!...!.!!..!
!!!..!...!.!.!!.!!!..!.!..!.
.!!!.!!..!..!..!!.!!!!..!....!!.!!..!...!.!.!..!.!.!!!!!!...!!!.!.....
!.!.!.!....!....!!..!!!!!.!.!!.....!.!!.!.......!......!!..!...!!!!!
..!.!!!.....!....!!!..!...!...!!...!....!..!!!...!!.!!!.!..!.!!....!!.
!!!!!!!....!.!...!!.!!!!.!.!!...!.!!.!...!.!.!.!..!.!.!!..!...!!.....!
....!!.!!..!!...!!..!!..!!..!.!!!!......!!.!.!..!..!..!..!...!...!!..!
!.!.!!.!!!!.....!.!!!..!!..!.!.!!....!!.!!!..!!.!.....!.........!.!..!
..!..!!.!!!...!!..!!!!..!!.....!!!.!...!!..!!!!!.....!.!..!...!.!!!!.!
!!.!.!..!....!!..!.!.!.!..!.!.!!!!.!!!..!!!..!!.!..!........!.!..!.!!!!
!..!.!...!!....!!!.!......!....!..!.!.!..!!.!!!!!!!..!..!!.!!!!...!...
!.!.!.!!.!!!!........!!....!....!!!....!!...!!.!....!!!...!..!...!..!!
!...!.!!!.!!.!!..!..!!!...!!!!.!!!..!..!!!..!!..!!!!!.!!!.!!!.!!!.!.!!
.!!!!!.!..!..!.!.!!!!!.!!!..!!.!!!!..!.!!.!.!!.!!.!!......!.!!!....!!.
.!!!..!...!!!!!..!!!...!!!..!!!..!..!!..!!..!.!.!.!.!!.!....!..!...!..
!.!!.!!!!!!!!....!.....!!...!...!.!!.!!..!!......!.!...!!..!!.!..!!!!!
..!.!!..!!.!.!..!.!!!!..!!..!!.!!..
Success rate is 47 percent (1346/2835), round-trip min/avg/max = 168/219/784 ms
R18#
```

FIGURE 16. Enormous packet loss due to heavy load on link.

Experimentally, we notice in Fig. 15 and Fig. 16, high load on a link can cause almost half the packet to be dropped and increase in the congestion levels on the router which eventually lead to possible device failure and high power consumption. Alternatively, we can notice that about 47G for this case was dropped and re-routed which indicate an efficient link management and less power consumption per device.

VII. CONCLUSIONS AND FUTURE WORK

Software Defined Networks (SDN) with the implementation of open flow allows to randomly innovate and validate network algorithms for the control layer. The work in this paper explorer the usefulness of developing a routing protocol that can affect the hybrid-SDN architecture significantly. The proposed algorithm, called iPRDR addresses the process of reducing energy consumption levels which is a vital factor in many production networks. iPRDR provides the functionality of preventing link congestion and reduce overload traffic avoiding the drawback in traditional non-SDN architecture by dynamically detecting the big traffic and shifting the flow to the appropriate processing unit, thus preventing any fails that may occur. Extensive experimental testing showed that the algorithm behaves as expected, solving the problem of link and cpu congestions of the routers by reducing 8.3% of big data traffic which is equivalent to 0.85kw/day. Consecutively, reducing the power consumption levels that are associated with the amount of traffic that is being processed per second by 34.9% of the operational power. In addition, the iPRDR implementation in the hybrid SDN environment allowed us to examine a realistic medium scale testbed environment rather than relying on simulation tools which may not emulate the realistic data centre operation. Future work on the iPRDR will be focused on implementing it in a large scale environment within a cloud- based SDN to manage and control multiple traditional data centre site locations with more advance power optimization features. The performance of the system will be assessed based on the complex topology and to discover any challenging limitations for the algorithm.

REFERENCES

[1] M. N. Siraj, N. Javaid, Q. Shafi, Z. Ahmed, U. Qasim, and Z. A. Khan, "Energy aware dynamic routing using SDN for a campus network," in *Proc. 19th Int. Conf. Netw.-Based Inf. Syst.*, 2016, pp. 226–230.

[2] H. Xu, X. Y. Li, L. Huang, H. Deng, H. Huang, and H. Wang, "Incremental deployment and throughput maximization routing for a hybrid SDN," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1861–1875, Jun. 2017.

[3] M. Canini, A. Feldmann, D. Levin, F. Schaffert, and S. Schmid, "Software-defined networks: Incremental deployment with panopticon," *Computer*, vol. 47, no. 11, pp. 56–60, Nov. 2014.

[4] Cisco. (2008). *Data Center Multi-Tier Model Design—Cisco*. Accessed: Aug. 20, 2017. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_2.html

[5] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An energy saving routing algorithm for a green OSPF protocol," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

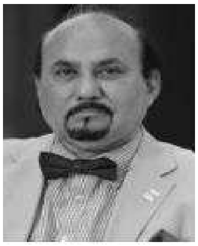
[6] L. M. Correia *et al.*, "Challenges and enabling technologies for energy aware mobile radio networks," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 66–72, Nov. 2010.

[7] B. Mumey, J. Tang, and S. Hashimoto, "Enabling green networking with a power down approach," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 2867–2871.

[8] A. Azari and G. Miao, "Network lifetime maximization for cellular-based M2M networks," *IEEE Access*, vol. 5, pp. 18927–18940, 2017.

[9] Z. Zhou, C. Gao, C. Xu, T. Chen, D. Zhang, and S. Mumtaz, "Energy-efficient stable matching for resource allocation in energy harvesting-based device-to-device communications," *IEEE Access*, vol. 5, pp. 15184–15196, 2017.
[10] F. Giroire, J. Moulierac, and T. K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 2523–2529.
[11] L. Prete, F. Farina, M. Campanella, and A. Biancini, "Energy efficient minimum spanning tree in OpenFlow networks," in *Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, Oct. 2012, pp. 36–41.
[12] D. Levin and M. Canini, "Panopticon: Reaping the benefits of incremental SDN deployment in enterprise networks," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2014, pp. 333–345.
[13] K. Xie, X. Huang, S. Hao, M. Ma, P. Zhang, and D. Hu, "E^3MC: Improving energy efficiency via elastic multi-controller SDN in data center networks," *IEEE Access*, vol. 4, pp. 6780–6791, 2016.
[14] S. Vissicchio *et al.*, "Safe updates of hybrid SDN networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1649–1662, Jun. 2013.
[15] K. Ohsugi, J. Takemasa, Y. Koizumi, T. Hasegawa, and I. Psaras, "Power consumption model of NDN-based multicore software router based on detailed protocol analysis," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1631–1644, May 2016.
[16] OpenNetworking.org. *Software-Defined Networking (SDN) Definition—Open Networking Foundation*. Accessed: Aug. 25, 2017. [Online]. Available: https://www.opennetworking.org/sdn-definition/
[17] P. Goransson, C. Black, and T. Culver, "Software defined networks," in *A Comprehensive Approach*, 1st ed. New York, NY, USA: Elsevier, 2014, pp. 59–79.
[18] (2014). *Cisco Open Network Environment: Bring the Network Closer to Applications*. Accessed: Sep. 20, 2017. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-1000v-switch-vmware-vsphere/white_paper_c11-728045.html
[19] C. H. W. Odom, *Foundation Topics >IPv4 Routing Protocol Concepts*. Indianapolis, IN, USA: Cisco Press, 2014, accessed: Sep. 8, 2017. [Online]. Available: http://www.ciscopress.com/articles/article.asp?p=2262897&seqNum=2
[20] (2016). *Cisco Data Center Spine-and-Leaf Architecture: Design Overview White Paper*. Accessed: Aug. 12, 2017. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white-paper-c11-737022.html
[21] opendaylight.org. *Platform Overview—OpenDaylight*. Accessed: Sep. 10, 2017. [Online]. Available: https://www.opendaylight.org/what-we-do/odl-platform-overview

AMMAR K. AL MHDAWI is currently pursuing the Ph.D. degree in electronic and computer engineering with Brunel University London, U.K. He is also a Cisco Certified Professional Engineer with over ten years of experience in network engineering and telecommunications. His research interests include IoT, software-defined networks, wireless sensor networks, delayed tolerant networks, and renewable energy.



HAMED S. AL-RAWESHIDY (M'92–SM'97) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the Post Graduate Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991. He was with: the Space and Astronomy Research Centre, Baghdad; PerkinElmer, Waltham, MA, USA; British Telecom, London, U.K.; Oxford University, Oxford, U.K.; Manchester Metropolitan University, Manchester, U.K.; and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network Communications Centre, Brunel University London, London.

• • •