IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 7, 2021, accepted June 7, 2021, date of publication June 14, 2021, date of current version June 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3089020

Soft-Mesh: A Robust Routing Architecture for Hybrid SDN and Wireless Mesh Networks

MUKHTIAR BANO[1], (Member, IEEE), AMIR QAYYUM[1], (Senior Member, IEEE), RAO NAVEED BIN RAIS[2], (Member, IEEE), AND SYED SHERJEEL A. GILANI[1]
[1]Faculty of Engineering, Capital University of Science and Technology (CUST), Islamabad 44000, Pakistan
[2]College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates

Corresponding author: Mukhtiar Bano (mukhtiar.bano@fjwu.edu.pk)

This work was supported by the Deanship of Graduate Studies and Research (DGSR) Program, Ajman University, United Arab Emirates.

ABSTRACT Wireless Mesh Networks (WMNs) are considered self-organizing, self-healing, and self-configuring networks. Despite these exciting features, WMNs face several routing challenges including scalability, reliability and link failures, mobility, flexibility, and other network management issues. To address these challenges, WMNs need to make programmable to allow modifications of standard techniques to be configured and implemented through software programs that can be resolved by integrating Software Defined Networking (SDN) architecture. SDN, being a cutting-edge technology promises the facilitation of network management as well as routing issues of wireless mesh networks. However, the evolution of the legacy IP-based network model in its entirety leads to technical, operational, and economic problems that can be mitigated by full interoperability between SDN and existing IP devices. This study introduces a Robust Routing Architecture for Hybrid Software-Defined and Wireless Mesh Networks (Soft-Mesh), by systematic and gradual transitioning of WMNs to SDNs in an efficient manner. The main objective of this paper is to suggest improvements to the architecture of the SDN node that allow the implementation of various network functions such as routing, load balancing, network control, and traffic engineering for the hybrid SDN and IP networks. Mininet-WiFi Simulator is used to perform various experiments to evaluate the performance of proposed architecture by creating a hybrid network topology with a varying number of nodes that is 50, 100, 150, 200, and 250 including SDN hybrid and legacy nodes with varying proportion of SDN hybrid and legacy nodes. Results are taken for the average UDP throughput, end-to-end delay, packet drop ratio, and routing overhead while comparing with traditional routing protocols including Optimized Link State Routing (OLSR) and Better Approach to Mobile Adhoc Networking (BATMAN) and with existing hybrid SDN/IP routing architectures including Hakiri and wmSDN. The analysis of simulation results shows that the proposed architecture Soft-Mesh outperforms in terms of the aforementioned performance metrics than the traditional and exiting hybrid routing protocols. Soft-Mesh gives 50% to 70% improved results concerning the incremental proportion of SDN hybrid nodes.

INDEX TERMS Control plane, data plane, hybrid, programmable, routing, software-defined networking, wireless mesh networks.

I. INTRODUCTION

The rapidly growing user demand and network usage make the management of traditional networks more complex and difficult to control. Although various types of network traffic and applications, such as multimedia, mobile data, cloud computing, and large data applications, have been used to generate high revenue, these applications still pose many operational and performance challenges for network operators [1]. While addressing these challenges, efficiency and flexibility remain key requirements for contemporary networks, however, network programming is a means of making these networks more efficient and flexible. Network programmability can be achieved by using the Software-Defined

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

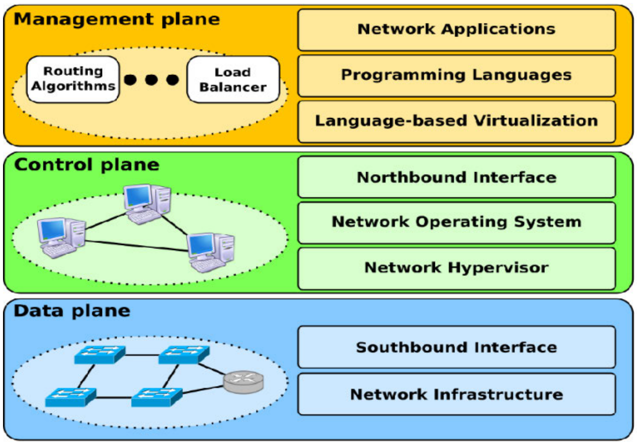
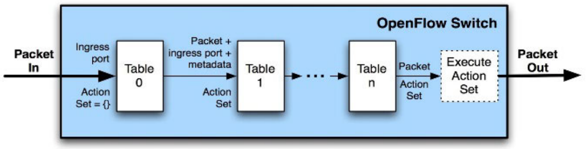


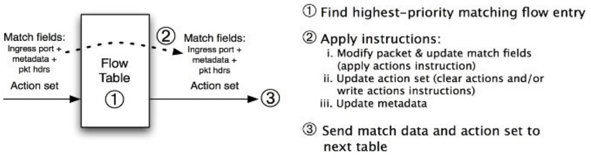
FIGURE 1. Software-defined networking (SDN) Architecture.

Networking (SDN) paradigm, which works on the principle of centralization of control and management, providing a solution for network management and control-related issues. SDN decouples the data plane from the control plane [2] and [3] and ensures a significant reduction in the complexity of network management, enabling revolutionary innovation and transformation with network interface programmability [4], [5], and [6]. The SDN architecture is advantageous over traditional network architectures in a variety of ways [7] and [8] including modification of traffic engineering policies, optimization of online or run-time traffic, creating innovative services like packets are treated differently based on the user or the application. SDN combines with Network Function Virtualization (NFV) for dynamic start-up and on-the-fly deployment of network functions and services, enabling network operators and service providers to gain control over their network.

As shown in Figure 1, the SDN architecture usually consists of three main layers, the application layer, control plane, and data plane. The network functions and applications commonly used by the organizations, such as security systems, firewalls, and load balancing, are included in the application layer. A layer of the control plane, known as the SDN brain, represents the logically centralized SDN controller. The controller administers the policies while residing on a computer. Physical devices such as routers, nodes, and other network devices collectively create the data plane layer. Using applications programming interfaces (APIs), known as northbound and southbound APIs, three-layer communication is carried out. In particular, to communicate with the controller, an application uses northbound API such as RESTful, while southbound API such as OpenFlow is used for communication between a controller and data plane devices. OpenDaylight, Floodlight, Ryu, and POX are commonly used controllers [9]. Whenever the first packet of a flow is received by the SDN node (implemented with the OpenFlow protocol), the node then queries the controller to get the forward path for the received flow, the controller installs the appropriate rules in the node firmware. As shown in Figure 2, these rules provide details about the behavior needed by the packet. The role of the controller is to install the necessary rules for each node involved in the forwarding path. A route request message is sometimes initiated from the originating node that does not have a path installed related to the corresponding packet.



(a) Packets are matched against multiple tables in the pipeline



(b) Per-table packet processing

FIGURE 2. Working of SDN.

The existing research on SDN is primarily focused on a complete paradigm shift from legacy networks to software-defined networking, considering the extensions and modifications in OpenFlow protocol. However, the current research is focused on the integration of SDN technology in the WMN networking paradigm to create a hybrid (SDN/IP) network architecture. As the architecture of Wireless Mesh Networks (WMNs) is increasingly implemented in existing communication systems and internet access applications due to its stability [10]. However, complex topology changes and diverse communication may be needed for the interaction of mesh nodes with the network, which is a major reason for making it difficult to manage WMNs [11]. Furthermore, effective load balancing, traffic engineering, and resource allocation must be introduced by the WMNs to alleviate these issues [12], [13], and [14]. In addition to the mentioned network management requirements, WMNs face several routing challenges including variable link quality, network heterogeneity, and traffic load [15], [16], and [17]. Being a new and promising architecture, SDN addresses the afore-mentioned routing and management challenges of WMNs by introducing agility and Flexibility [18]. SDN accomplishes the purpose of programmable WMNs by remotely controlling and configuring mesh routers and making them simple data forwarders. Furthermore, it is possible to implement congestion control and load balancing strategies to improve traffic management in WMNs. However, the adoption of SDN using a single controller may cause a compromise in network reliability issues. Fault tolerance must also be addressed when altering the SDN paradigm [19]. Considering user and system specifications, different QoS policies must be implemented by network operators [20]. Besides, the introduction of SDN in the WMN provides an efficient means of performing traffic engineering, resulting in a substantial improvement in network efficiency [21]. However, for the widespread implementation of any modern networking technology, Rip-and-replace is not a feasible technique. When adapting to a modern technology model, network operators are not expected to update forklifts [22]. Because of certain economic and operational problems, such as support for the entire network nodes and managed by a logically centralized controller, the full implementation of SDN does not work. Hence, incremental and hybrid deployment of SDN in WMN is necessary [23].

In addition, networks are becoming increasingly sensitive to the data handling, speed, and processing capabilities of the wireless nodes. Furthermore, the issues occur when SDN routing is implemented along with existing protocols. This results in multi-hop link-layer routing shortcomings, such as MAC layer-based routing, where a limited number of wireless nodes are enabled in a single network [24]. The implementation of OpenFlow nodes, instead of legacy nodes, gives rise to versatility in deploying data processing functionality, such as filtering or routing, which can be handled using several protocols. This versatility would encourage progress in the areas of mobility management, advanced routing, and traffic engineering and, more generally, in the optimization of the use of restricted WMN communication resources. As the logic for network control runs on a centralized controller tasked with matching criteria and processing actions for the OpenFlow network nodes, WMN-integrated OpenFlow simplifies network management. To achieve these benefits, we must overcome some WMN-related challenges, such as the unreliability of radio channels, which can disrupt controller communications temporarily, or the lack of layer 2 switching mechanisms like Spanning Tree or Auto Learning, which are commonly used to aid node and controller communications in wired deployment.

This paper is an extended version of our conference paper SDNHybridMesh: A Hybrid Routing Architecture for SDN Based Wireless Mesh Networks, published in Proceedings of the Web, Artificial Intelligence and Network Applications (WAINA 2020), Advances in Intelligent Systems and Computing, vol 1150, Springer, Cham [25]. We are making this version available to have more clear results and discussions in comparison to its short version. We propose Soft-Mesh, a robust routing architecture for hybrid SDN and Wireless Mesh Networks, to alleviate SDN implementation challenges and to achieve seamless interoperability between SDN and legacy nodes. Soft-Mesh modifies the SDN node architecture by making it hybrid and cohabitating with IP-based forwarding with OLSR routing and SDN forwarding with OpenFlow protocol. It should be noted that our research does not seek to support the replacement of traditional routing approaches by the SDN approach. The goal of this work is to investigate whether an architecture based on SDN can assist WMN routing and not arbitrarily equate legacy routing protocols that use in-band signaling and a centralized approach that uses out-of-band signaling. Considering legacy protocols, it is discussed how the SDN networking architecture can be used and to what degree the former can assist the latter. Moreover, Soft-Mesh architecture provides a cost-effective solution and seamless interoperability between legacy nodes and SDN nodes as compared to other hybrid routing architectures. The main contributions of this paper are:

i. A robust routing architecture for hybrid topology based on SDN hybrid nodes and legacy nodes that addresses the SDN controller's dynamic configurations to respond effectively to the network's topological changes, while considering the mobility of mesh nodes.
ii. The architectural modification of SDN nodes to create an SDN hybrid node that enables the data plane to react to changes in the network topology without requiring the controller to query each time when flows are inserted into the network.
iii. Classification of hybrid (SDN/IP) routing schemes as coexistence-based and cohabitation-based.
iv. Adaptive network monitoring module to mitigate the challenge of congestion control.
v. Comparison of performance metrics including the average UDP throughput, end-to-end delay, packet drop ratio, and routing overhead between the proposed routing architecture and existing traditional and hybrid SDN/IP routing architectures, simulations using Mininet-Wifi have been performed.

The remainder of the paper is structured as Section II offers an overview of current SDN enabled WMN solutions, Section III describes the architectural design of Soft-Mesh, Section IV presents implementation details, Section V presents the simulation model for Soft-Mesh, Section VI discusses simulation results and their interpretations, and the paper is concluded by Section VII.

II. RELATED WORK

This section provides a comprehensive review of various hybrid routing schemes that leverage SDN architecture with WMNs to address the challenges of load balancing, traffic engineering, and mobility management generally raised by topological changes that mesh networks experience. Besides the routing schemes, this section also provides a comprehensive review of network monitoring APIs that have been researched so far. Considering the implementation frameworks used in hybrid routing schemes, we categorize these schemes as 1) Coexistence-based hybrid (SDN/IP) routing, and 2) Cohabitation-based hybrid (SDN/IP) routing.

A. COEXISTENCE-BASED HYBRID (SDN/IP) ROUTING

The aim of coexistence-based hybrid (SDN/IP) routing schemes is the implementation of a network topology-based SDN architecture in which topology combines SDN nodes and legacy nodes, as shown in Figure 3. The OpenFlow protocol is implemented in SDN nodes, while legacy routing protocols such as AODV, OSPF, OLSR, etc. are implemented in legacy nodes.

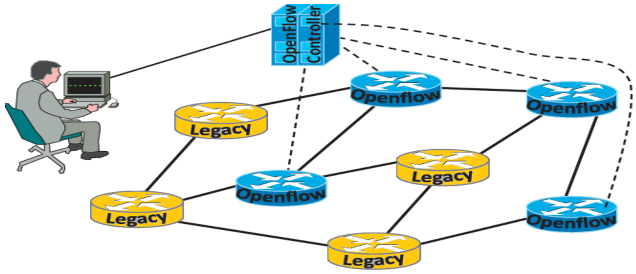


FIGURE 3. Coexistence-based hybrid (SDN/IP) routing.

A hybrid SDN architecture is suggested by Panopticon [26], which focuses on resolving interconnectivity issues between legacy and SDN nodes. In this architecture, network layer transmission issues are not considered. The different types of hybrid SDN models proposed by Vissicchio *et al.* [27] are categorized according to service type, class, and topology. Restricted research is done because there is no emphasis on individual implementation. HRFA [28] recommends a gradual introduction of SDN nodes by systematically increasing the number of nodes, which results in enhanced traffic forwarding. The proposed architecture is based on the OSPF [29] and OpenFlow protocols [30], though considering the use of links and the number of simulation efficiency metrics for hops. It achieves a huge reduction in congestion and load balancing of the network. Guo *et al.* [31] suggest a hybrid architecture that focuses on migrating from conventional to SDN paradigms, considering traffic engineering as a use case. This scheme uses a genetic algorithm to identify the migration sequence. Labraoui *et al.* [32] introduces a hybrid routing architecture with OLSR using SDN, aimed at researching the better performance and management of legacy routing supported by the controller. Reliability is often increased, but it affects network overhead, which rises linearly due to the increase in size of the network. The ratio of throughput and packet distribution is also increased. Wang *et al.* [33] consider a multi-hop wireless network integrated with SDN using residual energy and hop count as efficiency indicators for route selection, proposes an architecture addressing QoS Routing. To optimize broadcasts, the notion of Multipoint Relay (MPR) is applied. The shortest path is determined by the controller and multiple paths are obtained using the Dijkstra Algorithm. Another hybrid architecture is described in HEATE [34] that addresses the challenge of traffic engineering while considering energy efficiency. The architecture is based on the shortest path routing and OSPF protocol. It introduces a division of traffic flow, while traffic flows are aggregated to save electricity. He *et al.* [35] proposes an architecture focused on the deployment of the network zone, the ground is an IP network whereas space is the SDN network zone. Table 1(A) offers a description of hybrid routing schemes based on the coexistence of hybrid (SDN/IP) routing schemes, in which [26], [27] and [31] consider the physical position of the controller, the number of SDN nodes and the scalability of the controller as the most important control management challenges [36]. In wide or highly complex networks where controllers have to make fast decisions on a high frequency of events such as connection failures, dynamic traffic demands, regular arrival of new flows, etc., these problems may be very critical. However, [28] and [35] consider the sharing of topological information between traditional routers and SDN nodes as the most critical issues, and SDN nodes need to be intelligent enough to exchange link-state messages to get their neighbor's information.

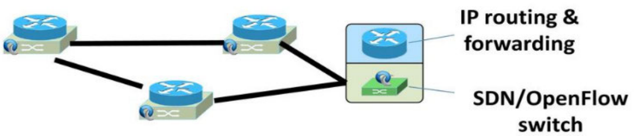


FIGURE 4. Cohabitation-based hybrid (SDN/IP) routing.

B. COHABITATION-BASED HYBRID (SDN/IP) ROUTING

The aim of cohabitation-based hybrid (SDN/IP) routing schemes is to modify the logical architecture of SDN hybrid node allowing cohabitation of OpenFlow and IP forwarding implemented with legacy routing protocol OLSR as shown by Figure 4. The former is to communicate with SDN nodes, and the latter is for legacy nodes communication.

Mesh Flow [37], a node-based hybrid architecture is proposed to address various hybrid SDN problems such as performance enhancement, efficient and scalable customer mobility, scalable routing, load balancing, etc. Its physical interface is split into virtual interfaces to which a specific SSID is allocated. To support data and to control traffic management, these virtual interfaces are used. This architecture is, however, badly impacted by topological shifts. OLSR is used [38] to regulate traffic. The centralized controller is used to manage the traffic of data using OpenFlow, and the allocation of resources is also configured. The architecture uses a monitoring and control manager that assists in the management of mobility and the NOX operating system that produces flow tables. wmSDN [39] uses Mesh Flow as a reference architecture and some changes were introduced as a single point of failure when addressing the controller's main challenge. As a backup, it utilizes a distributed control mechanism. A single SSID for traffic data and control is used. The architecture uses mesh access points (MAP) that are linked to the centralized controller.

Also, each MAP consists of various virtual interfaces that connect to other nodes. Data and control management is done by using various subnets. OLSR takes care of routing in this architecture in case of controller failure and manipulates the switching table. Such architecture achieves traffic optimization. Salsano *et al.* [40] suggest another hybrid architecture, which is based on the extension of wmSDN to enhance fault tolerance. It utilizes several Embedded Flow Table Manager (EFTM) controllers and MAPs; flow tables and controllers are designed accordingly. Synchronization between controllers is a major challenge facing such architecture [41].

TABLE 1. **Existing hybrid routing schemes.**

Architecture	**Features**	**SDN Routing Protocol**	**IP Routing Protocol**	**Challenges**
A. COEXISTENCE-BASED HYBRID (SDN/IP) ROUTING SCHEMES				
Panopticon [26]	Load balancing, fault tolerance, link recovery	Open -Flow	Shortest path routing	Location and number of SDN switches
Vissicchio [27]	Flexibility, partial robustness	Open -Flow	Shortest path routing	Topological position and scalability
HRFA [28]	Load balancing, congestion control, and making traffic fast-forwarding	Open -Flow	OSPF	Information exchange b/w SDN and IP nodes
Guo [31]	Traffic Engineering and link utilization	Open -Flow	OSPF	Migration sequence of SDN nodes
Labraoui [32]	Reliability and performance improvement	Open -Flow	OLSR	One hop communication b/w Controller and nodes
Wang [33]	QoS Routing, Single/multipath routing, and disjoint multipath	Open -Flow	OLSR	Expansion in network size
HEATE [34]	Traffic engineering, Energy saving	Open -Flow	OSPF	Energy consumption by link,
He [35]	Collaborative management,	Open -Flow	RIP and OSPF	Information exchange b/w SDN and IP nodes
B. COHABITATION-BASED HYBRID (SDN/IP) ROUTING SCHEMES				
Mesh Flow [37]	Load balancing, mobility management, and optimization of resource allocation	OLSR	Open-Flow	Fault tolerance
wmSDN [39]	Reliability under controller failure and traffic optimization	OLSR	Open-Flow	Excessive packet-in traffic, dynamic topology
Multi-Controller Mesh [40]	Multiple controllers' deployment, fault tolerance	OLSR	Open-Flow	Synchronization of all active controllers
OSHI [42]	A management tool, fast restoration, and traffic engineering	Open-Flow	MPLS	Frequent rule updating, Dynamic topology

An open-source architecture that combines the OpenFlow node, IP routing daemon, and IP engine is OSHI [42]. Quagga is used as an IP routing daemon [43] and [59], for best-effort IP and SDN routing, MPLS labels are used. In addition, all the control plane functions are implemented by using SDN. Hakiri *et al.* [44] use SDNs introduced with WMNs to incorporate network virtualization mechanisms, routing, and traffic engineering in smart cities, enabling an increase in network capacity and flexibility. Table 1(B) summarizes cohabitation-based hybrid routing schemes, in which [37], [39], and [42] consider fault tolerance, excessive control traffic, and dynamic topology as the most significant challenges for the coexistence of SDN and IP in one node [45]. These challenges may be mitigated by designing SDN nodes in such a way as to have a local management entity that may be used to create a logical interface between the two different paradigms of centralized and distributed network solutions [46]. Shastry and Kumar [47] presents software-defined wireless mesh network architecture to address traffic balancing issues caused by node mobility. The proposed model estimates the probability of connection failure in the topology to reduce the overall response time of the SDN controller in the complex network topology. Once a connection failure is expected, an alternative set of different routes is proposed based on the successful stability of traffic in the network, reducing control plane overhead. Kuznetsova *et al.* [48] uses software-defined networking (SDN) to manage wireless mesh sensor networks, where the network management is done with the help of an SDN controller, thereby improving bandwidth, jitter time, and packet loss performance.

C. NETWORK MONITORING APPLICATIONS FOR HYBRID (SDN-WMN) ARCHITECTURES

OpenNetMon [49], a network monitoring module is proposed to monitor the network failure ratio, throughput, and delays of packets. The traffic matrix for each flow cannot be obtained due to limited Ternary Content-Addressable Memory (TCAM), a specialized high-speed memory using single clock cycle scans all its contents. iSTAMP [50] addresses this issue, which seeks a balance between the limitations of network resources and the precision of measurement using aggregation and de-aggregation mechanisms [51]. Its drawback involves ignoring constraints when aggregating flows and several times using the TCAM table for a single flow, thereby raising the cost of measurement [52]. OpenTM [53], uses traffic matrix estimation to detect active flows based on routing and forwarding route information supplied by the controller. RESTful uses PayLess framework [54] focused on low-cost flow calculation using polling to gather flow statistics. It suggests an adaptive approach for gathering statistics. FlowSense control module [55] measures changes that are dynamically occurring in network flows. By taking statistics

TABLE 2. Existing network monitoring applications.

Method	Features	Analysis
OpenNetMon [49]	Data fetching is adaptive	Increase in accuracy with increasing overhead
iSTAMP [50]	Partitioning of TCAM for traffic aggregation and de-aggregation	Increase in accuracy with additional mechanism prioritize flows
Moshref [51]	Hash-based switches for collection of traffic information	Increase in accuracy with the careful delegation of monitoring rules
Zhang [52]	The algorithm based on prediction to count flow and detect anomalies	Identified traffic gives more accuracy
OpenTM [53]	Continuous polling to collect flow statistics	Increase in accuracy with increasing overhead
Payless [54]	The polling algorithm is adaptive with flow frequency	Variation in accuracy and overhead with polling interval length
FlowSense [55]	PacketIn and FlowRemoved OpenFlow messages used	Increase in accuracy with decreasing overhead
DREAM [56]	Dynamic deployment of resources with the required level accuracy level	Concurrent tasks give more accuracy
SOFTmon [57]	NOS-independent monitoring with utilization information on switch, port, and on a flow level	Accurate for specific tasks only
Bao [58]	Prediction of node mobility and link failure probability by using a supervised learning model	Routing overhead increases with the speed of node

from control messages, it performs mathematical modeling to get throughput and other metrics. DREAM [56] provides the necessary level of accuracy for the dynamic deployment of resources, while more precision is provided by concurrent tasks. SOFTmon [57] proposes a NOS-independent monitoring system using a switch, port, and flow-level information, but accuracy is only for specific tasks. A summary of existing network monitoring applications used by SDN-enabled WMN is presented in Table 2.

III. ARCHITECTURAL DESIGN OF SOFT-MESH

When a node needs to connect with another node in the SDN networking model, it queries the controller to provide path information for that particular node, resulting in the implementation of the routing rules needed. Connection to the controller node plays an important role while establishing connectivity from one node to another in this regard. Although, due to the static existence of network nodes, routing rules do not change or update regularly on wired networks, resulting in no effect on the controller's node connectivity. But the regular movement of nodes in wireless networks particularly wireless mesh networks greatly impacts communication between nodes. Soft-Mesh architecture is based on backbone routing and network monitoring issues associated with SDN-enabled WMN. The topology under consideration comprises an SDN Controller linked to SDN hybrid nodes and legacy nodes as shown in Figure 5. OpenFlow for traffic management and IP-based forwarding for data packet transmission is used in this method. Routing, network monitoring, traffic measurement, and load balancing modules are part of the controller.

A. ROUTING MODULE

This module implements the shortest path algorithm to create an effective routing strategy to route packets through the hybrid nodes and legacy nodes [60] and [61]. The hybrid node routing function consists of two submodules, one supporting SDN routing with OpenFlow protocol implementation to communicate routing rules and policies, and the other supporting IP routing via the legacy OLSR routing protocol as shown by Figure 6. SDN/IP cohabitation is desirable in a way that the controller can provide the best path choices that are transmitted to nodes, if possible. However, in the event of the controller being unavailable or malfunctioning, the legacy routing protocol is used to send packets. Each SDN node keeps its neighbors identified, periodically creates a new refreshed routing table, and selects the shortest new route to all destinations. In this way, the controller retrieves topology information from its nearby SDN nodes.

B. NETWORK MONITORING MODULE

In general, flow statistics are collected by regularly polling the nodes with a pre-defined time interval by sending *FlowStatisticsRequest* control message. The polling frequency must be high to obtain precise statistics, but it will increase the monitoring overhead of the network [62] and [63]. An adaptive flow statistics processing algorithm is proposed to maintain a balance between the precision in the processing of statistics and network overhead. We suggest that when a *PacketIn* message is received by the controller, it adds a new flow entry to an active flow table along with an initial collection timeout for statistics, i.e., τ milliseconds.

The controller obtains its statistics in a *FlowRemoved* message if the flow expires within a few milliseconds. Otherwise, the controller will send a *FlowStatisticsRequest* message to the corresponding node in response to the timeout event after τ milliseconds, to collect statistics about that flow. If during this period the data collected for that flow does not change substantially, i.e., the difference between the previous and

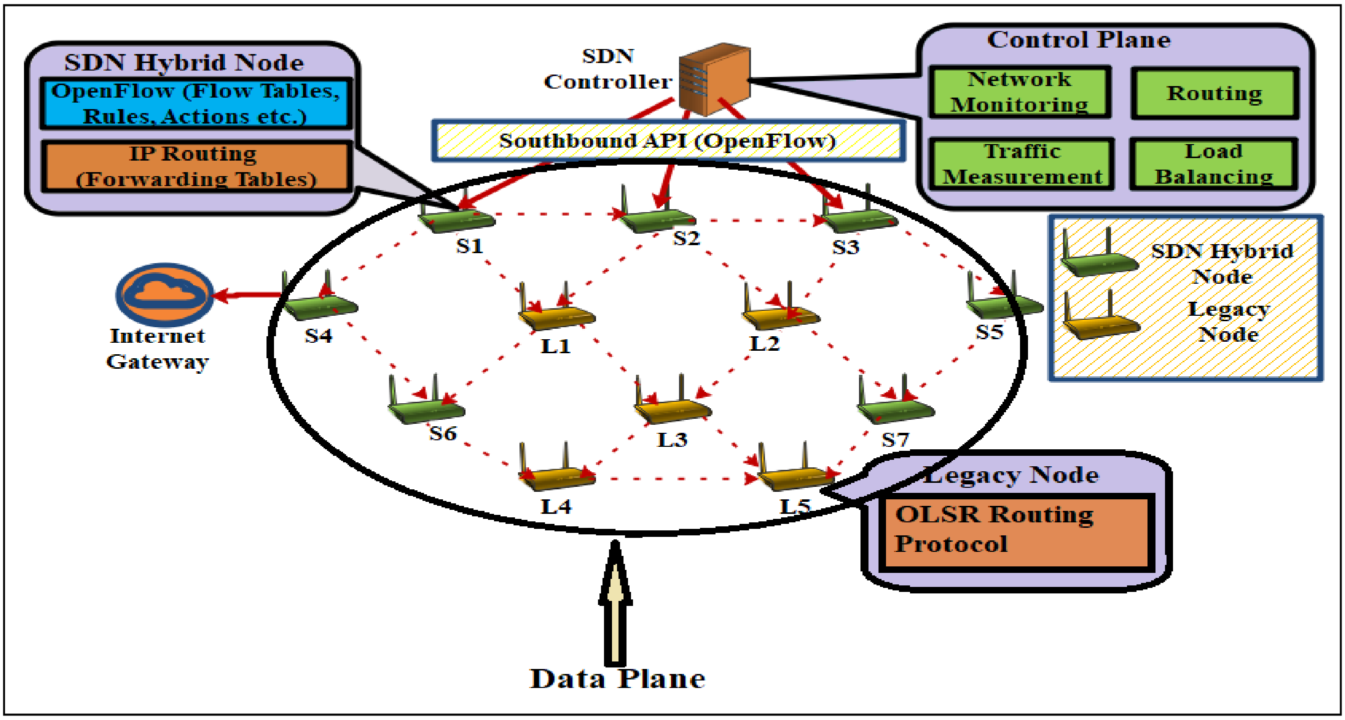


FIGURE 5. Architecture of soft-mesh.

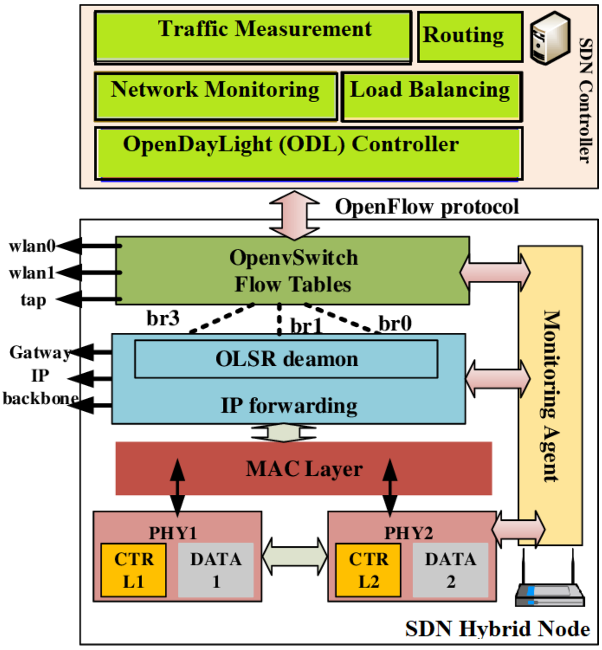


FIGURE 6. The SDN hybrid node Architecture.

current byte count against that flow is not above a threshold, say Δ_1, the timeout for that flow is multiplied by a small constant, say $\boldsymbol{\alpha}$. This process may be repeated until a maximum timeout value of $\mathbf{T_{max}}$ is reached for a flow with a low packet rate. If the difference in the old and new statistics becomes larger than another threshold Δ_2, the scheduling timeout of that flow is divided by another constant $\boldsymbol{\beta}$. This process may be repeated for a heavy flow until a minimum timeout value of $\mathbf{T_{min}}$ is reached. We maintain a higher polling frequency for flows that significantly contribute to link utilization, and a lower polling frequency for flows that do not significantly contribute towards link utilization at that moment. If their contribution increases, the scheduling timeout will adjust according to the proposed algorithm to adapt the polling frequency with the increase in traffic. This algorithm is further optimized by batching *FlowStatisticsRequest* messages together for flows with the same timeout resulting in a reduction of the spread of monitoring traffic in the network without affecting the effectiveness of polling with a variable frequency. Algorithm 1 demonstrates this algorithm's pseudocode, and Figure 7 illustrates the network monitoring module of Soft-Mesh.

C. TRAFFIC MEASUREMENT MODULE

The controller queries the last SDN node on the forwarding path for traffic calculation, and the counter returns the number of packets of each flow in the sampling interval, thus obtaining the throughput of the forwarding path. The controller sends probe messages to the forwarding path's data layers. These messages travel through all nodes along the path and eventually return to the controller, whereby measuring time differences, contact delays can be obtained.

D. LOAD BALANCING MODULE

To alleviate the congestion problem, this module is triggered by a network monitoring and traffic analysis module

Algorithm 1 Flow Statistics Collection

```
globals: active_flows //Currently Active Flows
         schedule_table //Associative table of active flows
                        // indexed by poll frequency
         U // Utilization Statistics. Output of this algorithm
ife is Initialization event then
  active_flows ← $\Phi$, schedule_table ← $\Phi$, U ← $\Phi$
end if
if e is a PacketIn event then
  $f \leftarrow$ (e.switch, e.port, $\tau_{\min}$, 0)
  schedule_table[$\tau_{\min}$] ← schedule_table[$\tau_{\min}$] $\cup f$
else if e is timeout $\tau$ in schedule_table then
  for all flows $f \in$ schedule_table[$\tau$] do
    send a FlowStatisticsRequest to f.switch
  end for
else if e is a FlowStatisticsReply event for flow f then
  diff_byte_count ← e.byte_count − f.byte_count
  diff_duration ← e.duration − f.duration
  checkpoint ← current_time_stamp
  U[f.port][f.switch][checkpoint] ← (diff_byte_count, diff_duration)
  if diff_byte_count < $\Delta_1$ then
    $f.\tau \leftarrow \min(f.\tau\alpha; \tau_{\max})$
    Move f to schedule_table[f. $\tau$ ]
  else if diff byte count > $\Delta_2$ then
    $f.\tau \leftarrow \max(f.\tau/\beta, \tau_{\min})$
    Move f to schedule_table[f.$\tau$ ]
  end if
end if
```

Algorithm 2 Load Balancing Algorithm

```
rules ← DefaultRules();
trafficSchudeling();
while Listening to LLDP packets do
  isOptimalPATH = optimal_path(rules);
  if ≠ isOptimalPATH then
    rules ← calculateNewRules();
    FlowMod_router(); path ← optimalPath(rules);
  else
    installOFRules(path);
  end
  hostsReachable();
  monitoringPath();
end
```

after obtaining the node and connection statistics. Connection statistics such as bandwidth, latency/delay, and usage of connections are used by the controller and the traffic is routed to an appropriate link. To choose the optimal route, Algorithm 2 demonstrates the load balancing algorithm. It calculates the new rules for the new route to the new mesh nodes, i.e., the MAC and IP addresses. If the new path is formed by sending *FlowMod* messages end-to-end, the controller floods all ports to the selected virtual routers, opens the client link to allow packets to reach their destination, and continues to discover and monitor the topology of the network at the same time.

IV. IMPLEMENTATION DETAILS OF SOFT-MESH

In this section, implementation details of Soft-Mesh are presented. The SDN hybrid node architecture uses two tables, Figure 8 displays TCAM table and SRAM table respectively. The former is used for OpenFlow forwarding entries and the latter is used for IP forwarding entries. Each hybrid node forwards OpenFlow messages using the software router OpenVSwitch that implements a software pipeline based on flow tables [64]. Also included is an IP-based forwarding daemon running the standard OLSR routing protocol [65]. OpenVSwitch bridges OpenFlow and standard routing protocols by using virtual network interfaces to exploit IP networks' ability to route packets using the shortest path.

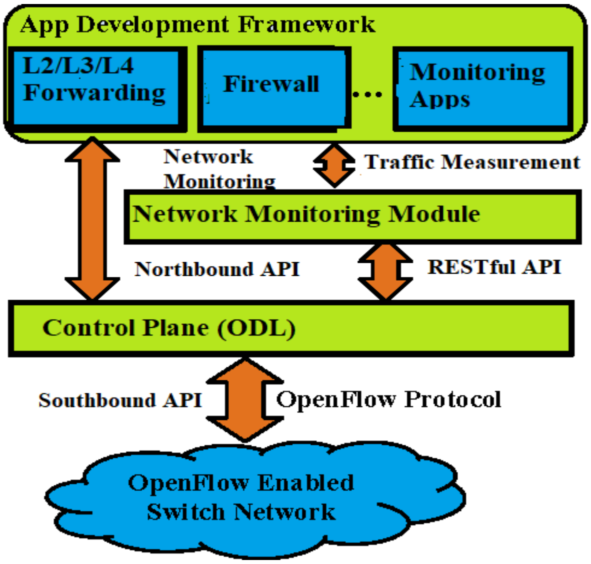


FIGURE 7. Soft-mesh network monitoring module.

When the first packet of a corresponding flow is received by a hybrid node, it processes the packet to determine the next hop to which it is to be forwarded, as shown in Figure 9. The following steps are carried out in the case of an OpenFlow message is received:

1. Extract the header for the packet
2. To balance the flow table, look at TCAM table entries
3. If the respective flow entry is not identified, then it forwards packet information to the controller.
4. The controller sends to the hybrid node the relevant rules
5. The SDN hybrid node flow table is updated accordingly
6. If the flow input in the TCAM table is found, the next hop is calculated, then the packet is forwarded.

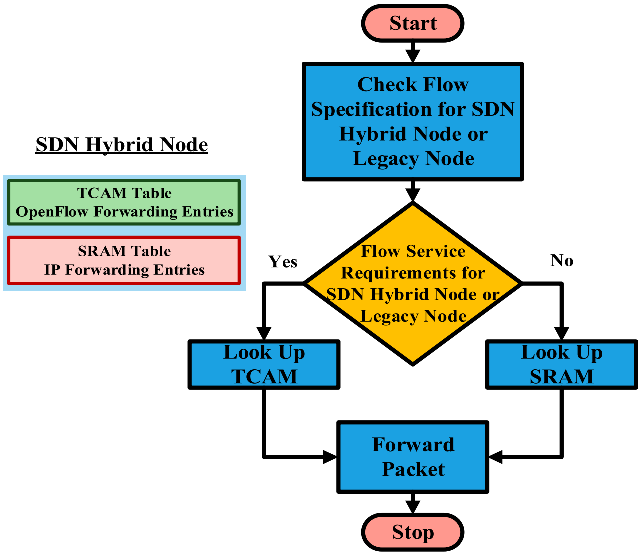


FIGURE 8. Working of SDN hybrid node.

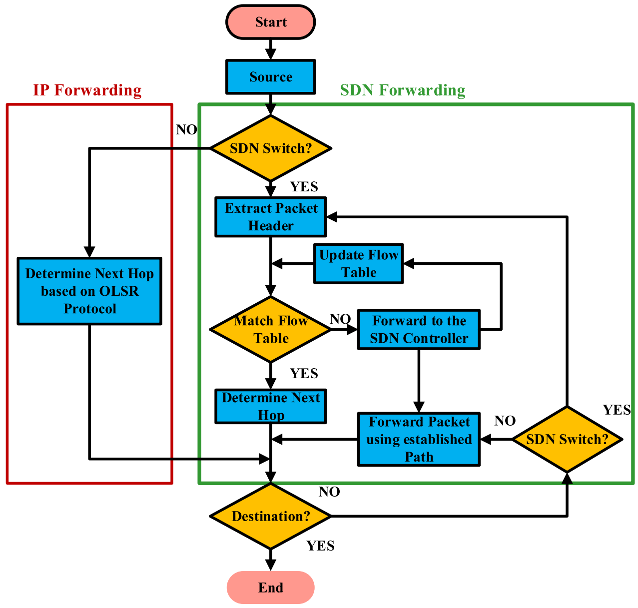


FIGURE 9. SDN hybrid node flow chart.

However, in case the IP packet is received, the next hop is calculated based on the OLSR routing protocol, and the packet is forwarded accordingly. Figure 10 illustrates how a legacy node operates, with the traditional OLSR routing protocol applied. It primarily gathers data from its 1-hop and 2-hop neighbors using HELLO messages, and then chooses Multipoint Relays (MPRs). To minimize the number of redundant retransmissions, multipoint relays are used when forwarding a transmitted packet. This strategy restricts the node collection retransmitted by a packet from all nodes to a subset of all nodes. In this manner, it computes its forwarding table. Moreover, Soft-Mesh makes use of Dijkstra Algorithm [66] for shortest path calculation.

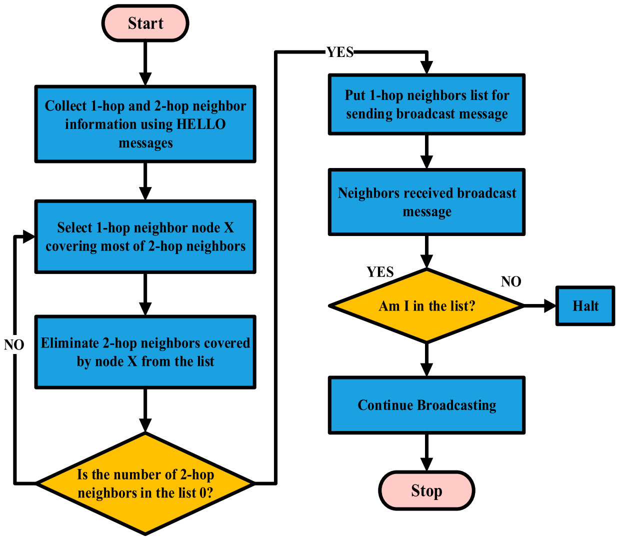


FIGURE 10. Working of legacy node.

A. MATHEMATICAL MODEL OF SOFT-MESH

This section presents the mathematical model of the proposed architecture which provides the basis for computing aggregated flows.

The graph of WMN is represented by

$$G = (V,E),$$

where V and E represent the set of all vertices (i.e., nodes) and edges (i.e., links), respectively.

Let TD_v denote the traffic demand of the flows generated in node v,

Let $TD_v = \{f^{\text{ID:1}}, f^{\text{ID:2}}, f^{\text{ID:3}}, \ldots, f^{\text{ID:n}}\}$ denote the traffic demands for n flows,

Given a traffic distribution Δ_{v}, the outgoing flows' demands in different paths of node v can be obtained by using the below model:

$$\{f^{\rho 1}{}_{\text{A1}}, f^{\rho 2}{}_{\text{A2}}, ..f^{\rho n}{}_{\text{An}}\} = \Delta_v \times TD_v \quad \forall_v \in V$$

where f^{ρ}_{A} implies the rate of flow A that is sent out from node v on path ρ.

As the aggregated flows must be less than their capacity $c(e)$ on each connection in the time slot τ, we have:

$$\sum\nolimits_{\rho \in f(e)} f^{\rho} \leq c(e) \quad \forall e \in E$$

The network monitoring and traffic measurement module makes use of aggregated flow parameters to get the statistics of network traffic on each node.

V. SIMULATION MODEL FOR SOFT-MESH

The deployment of topologies and the positioning of nodes in WMNs has become a difficult task. The neighborhood and interference relationships can vary depending on the placement [67] of nodes and controller. Soft-Mesh routing architecture considers the network topology based on SDN controller, SDN hybrid nodes, and legacy nodes. The physical placement of the SDN controller can be anywhere in the network, however, it must logically be centralized. We have assumed that the controller is placed at one hop distance with SDN hybrid nodes and legacy nodes. However, the SDN Hybrid node requires two interfaces and hence two subnets, one for the control packets and the other for the data packets. All nodes for the control subnet are directly accessible (one-hop) by the controller, whereas the data network is a standard multi-hop network where data traversed several routers to reach their destination. The controller would get the knowledge of other nodes that are more than one hop away by using OLSR information to reconstruct the full topology of the network for route calculation. This infers that a strictly centralized solution would be impossible and, combining SDN with a distributed routing protocol is used to deliver control messages to the controller from remote nodes.

Nodes periodically send their neighboring information tables to the controller using the control subnet, so the controller has a global view of the topology of the network. By using Djikstra 's algorithm, as shown by Algorithm 3, it periodically determines the shortest route (in terms of the number of hops) from the source to the destination. If an optimized path is identified, the controller informs participating nodes (source and intermediate routers) of the modified rules. Using the graph topology, which includes all available routers as well as the links connecting them, the controller calculates the new optimal path. Then, as demonstrated by Algorithm 4, it installs new OpenFlow rules to program flow entries within the software pipeline on each path. The nodes install the respective rules in their routing tables and the path is formed.

The implementation framework of Soft-Mesh is using Mininet-WiFi simulator, providing a simple and inexpensive network testbed for developing OpenFlow applications. The topology under study is based on a controller, legacy nodes along with SDN hybrid nodes. The OpenDaylight (ODL) SDN Controller [68] is used for this purpose and implemented Soft-Mesh on the ODL controller as a network application. We evaluated Soft-Mesh by comparing its performance with the traditional routing approaches OLSR and BATMAN [69], as these schemes are considered relatively more stable among all other traditional routing approaches for wireless mesh networks. However, for hybrid approaches, Soft-Mesh architecture has been compared with wmSDN [39] and Hakiri *et al.* [45], using performance metrics Average UDP Throughput, Packet Loss Ratio, End-to-End Delay, and Routing Overhead.

Experiments are performed for the topology consisting of a varying number of SDN hybrid nodes and legacy nodes that are 50, 100, 150, 200, and 250 nodes respectively. Moreover, the proportion of SDN hybrid nodes and legacy nodes is also variable to create three different simulation scenarios including 10%, 25%, and 50% of SDN hybrid nodes whereas 90%, 75%, 50% of legacy nodes in the network topology. The size of topology has also been kept variable that is the size of 500m*500m for 50 and 100 nodes topology and 1000m*1000m for 150, 200, and 250 nodes topology. To get more accurate results, experiments have been carried out several times that are approximately 10 experiments each for mentioned number of nodes have been performed while considering the Random walk mobility model to investigate support for node mobility in the algorithm. Moreover, to see the variations in results, standard deviations for the desired performance metrics have also been con-

Algorithm 3 Dijkstra Routing Algorithm

```
Input: $G$[a connected simple graph with a positive
           weight for every edge],
       $\infty$[a number greater than the sum of the weights
           of all the edges in the graph],
       $w(u, v)$ [the weight of edge {u, v}],
       a [the starting vertex],
       z [the ending vertex]
Algorithm Body:
Initialize $T$ to be the graph with vertex $a$ and no edges.
Let $V(T)$ be the set of vertices of $T$, and let $E(T)$ be the set
of edges of $T$.
    Let $L(a) = 0$, and for all vertices in $G$ except $a$
    Let $L(u) = \infty$
    [The number $L(x)$ is called the label of x.]
Initialize $v$ to equal $a$ and $F$ to be {a}.
    [The symbol v is used to denote the vertex most
    recently added to T.]
while ($z \notin V(T)$)
F: = (F — {v}) U {vertices that are adjacent to v
    and are not in $V(T)$}
[The set F is called the fringe. Each time a vertex is added
to T, it is removed from the fringe and the vertices adjacent
to it are added to the fringe if they are not already in the
fringe or the tree T]
    For each vertex $u$ that is adjacent to $v$ and is not
      in $V(T)$,
      if $L(v) + w(v,u) < L(u)$ then
      $L(u) := L(v) + w(v,u)$
      $D(u) := v$
[Note that adding v to T does not affect the labels of any
vertices in the fringe F except those adjacent to v. Also,
when $L(u)$ is changed to a smaller value, the notation $D(u)$
is introduced to keep track of which vertex in T gave rise
to the smaller value.]
Find a vertex $x$ in $F$ with the smallest label
Add vertex $x$ to $V(T)$, and add edge {D(x), x} to
    $E(T)$
    $v := x$ [This statement sets up the notation for the
    next iteration of the loop.]
end while
Output: L(z)
    [L(z), a nonnegative integer, is the length of
the shortest path from a to z.]
```

TABLE 3. Simulation parameters of soft-mesh Architecture.

PARAMETERS	VALUE
Simulator	Mininet-WiFi
Protocols	OLSR, BATMAN, wmSDN, Hakiri, Soft-Mesh
Simulation Time	100 seconds
PHY and MAC model	802.11n
No of nodes	50, 100, 150, 200, 250
Proportion (SDN hybrid nodes% + Legacy nodes%)	10%+90%, 25%+75%, 50%+50%
Topology Size	500m * 500m (50 and 100 nodes), 1000m * 1000m (150, 200 and 250 nodes)
Mobility Model	Random Walk
Mobility Speed	Min 10 m/s, Max 50 m/s
Traffic rate (Mbps)	5 Mbps
Traffic Type	UDP
Traffic flows Characteristics	Constant Bit Rate (CBR)
Packet Size (byte)	1024 byte
HELLO emission interval	2 seconds
Controller-Node message emission	3 seconds
Timeout for unused routes	10 seconds
Topology reconstruction interval	5 seconds

sidered. For MAC Layer, mac80211_hwsim is used, where operation time to start an AP is 17ms, the station starts at 63ms, two nodes associate at 10ms, AP and stations associate at 350ms. The traffic flows are of constant bitrate (CBR) type. The mobility model random walk follows the speed of the mobile node that is minimum of 10m/s and maximum 50m/s. The parameters used in our simulations are presented in Table 3.

VI. SIMULATION RESULTS AND DISCUSSION

This section presents the comparison of performance metrics including the average UDP throughput, end-to-end delay, packet drop ratio, and routing overhead between the proposed routing architecture and existing traditional and hybrid SDN/IP routing architectures. Table 4 describes the comparison of various features and differences among proposed architecture Soft-Mesh and existing traditional and hybrid (SDN/IP) routing approaches. Simulations are carried out for OLSR and BATMAN routing schemes as these schemes are considered relatively more stable among all other traditional routing approaches for wireless mesh networks.

A. THROUGHPUT

Figure 11 shows the average UDP throughput for each scenario, based on the number of network nodes including the proportion of SDN hybrid nodes and legacy nodes in the network topology. The statistics clearly show that the Soft-Mesh's centralized routing approach outperforms the distributed protocols as well as other hybrid approaches. This performance enhancement is directly linked to the inherent features of the SDN paradigm itself. The continuous monitoring of the topology of the network enables the rapid identification of changes in topology and the rapid recalculation of new routes for the network's ongoing traffic flows.

Algorithm 4 Optimized Path Algorithm

```
Data: rules, PATH
Result: Function to find optimal path optimalPath(rules);
if (∃ PATH in (rules) then
    PATH ← find(rules);
    return PATH
else
    rules ← calculateNewRules(); FlowMod_router();
    return rules
end
optimal_path(rules);
```

If this recalculation results in better identification of the router, the corresponding rules are pushed to the routers and mounted along the measured path without flooding the network. This contrasts with the distributed protocols in which network flooding is the primary mechanism for propagating topology change information. It is understood that this ''flooding'' process induces substantial delays and increases the convergence time of the routing protocols. In short, Soft-Mesh routing responds faster than traditional protocols to topology changes and guarantees optimal paths (in terms of hop count) at any given time, which in turn yields optimal performance in terms of UDP throughput. Using SDN with WMN, it is possible to fine-tune the flow distribution, such as load balancing between alternate routes.

B. PACKET DROP

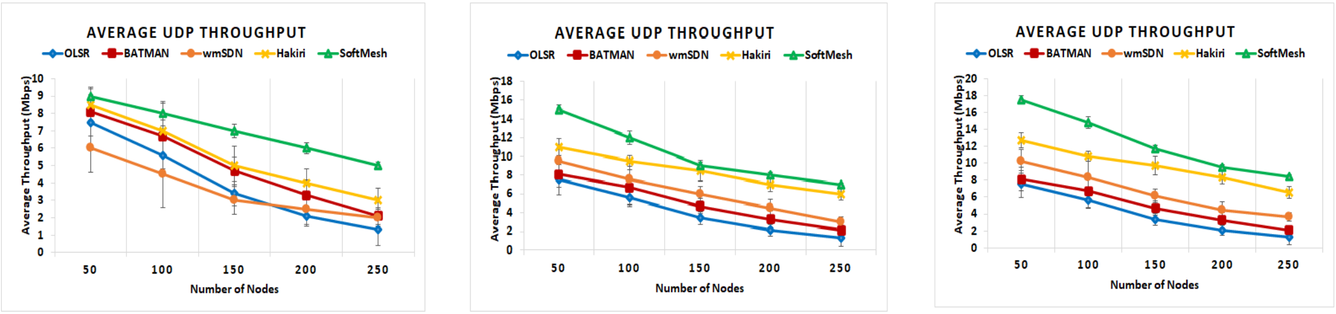
Packet loss is a good measure to represent the efficiency of a routing protocol in the optimization of internode interface exchanges. Due to the lack of a routing rule, data is collected about the packets lost. The packet drop ratio is shown by Figure 12, where OLSR has the largest dropped packet rate and therefore the slowest convergence phase, and BATMAN comes after due to a relatively long period between updates and the absence of an overhead optimization mechanism. BATMAN has less packet drop ratio than OLSR because of its buffering feature, packet loss increases with the number of nodes. As a consequence, the rate of packet loss will be directly influenced by two key parameters, which would be the ability to optimize routes via route protocols and the amount of overhead they incur. Nonetheless, in most situations, interference has a direct effect on the rate of packet loss due to collisions and wrong packets.

C. END-TO-END DELAY

Figure 13 shows the end-to-end delay as the time from a packet to be sent from the source mesh node before it is received by the destination node. We carried out this experiment several times and maintained the average latency. It is not easy to quantify one-way delays because packets encounter numerous network delays, including

TABLE 4. Comparison of soft-mesh with existing traditional and hybrid (SDN/IP) routing approaches.

Features	OLSR	BATMAN	wmSDN	Hakiri	Soft-Mesh
	Traditional Routing		**Hybrid (SDN/IP) Routing**		
Scalability	Easily Scalable	Easily Scalable	Possibly Scalable	Possibly Scalable	Easily Scalable
Reliability	High	High	Possibly High	Possibly High	Very High
Controller Failure Solution	No Controller exists	No Controller exists	Switch to traditional routing	No solution proposed	Switch to traditional routing
Resource Optimization	Not at all	Not at all	Possibly Optimized	Moderately Optimized	Highly optimized
Congestion Control Mechanism	Not at all	Not at all	No congestion control	Load balancing mechanism	Load balancing mechanism
Network Programmability	Not at all	Not at all	Programmable	Programmable	Programmable
Cost-effectiveness	Effective (All nodes are legacy nodes)	Effective (All nodes are legacy nodes)	Highly Expensive (All nodes need to be replaced with SDN hybrid nodes)	Highly Expensive (All nodes need to be replaced with SDN hybrid nodes)	Effective (Few nodes need to be replaced with SDN hybrid nodes)
Network Monitoring	Not at all	Not at all	Simple Network Monitoring	Simple Network Monitoring	Adaptive Network Monitoring

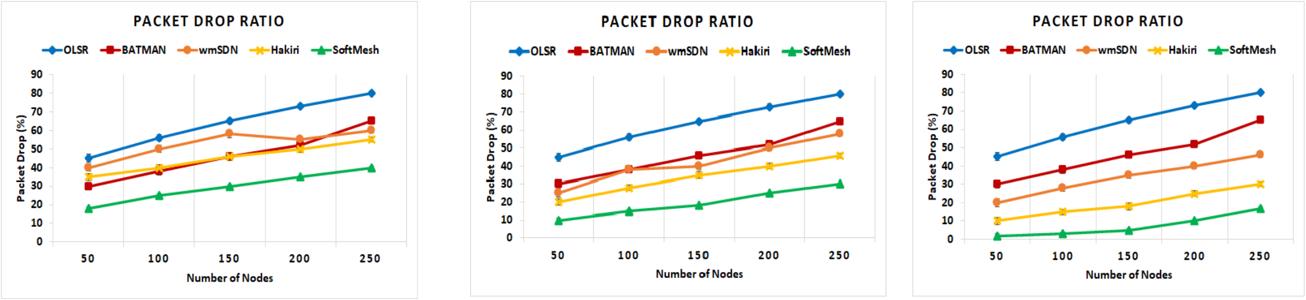


a) Scenario 01 (10% SDN hybrid nodes + 90% Legacy nodes)

b) Scenario 02 (25% SDN hybrid nodes + 75% Legacy nodes)

c) Scenario 03 (50% SDN hybrid nodes + 50% Legacy nodes)

FIGURE 11. Average UDP throughput.

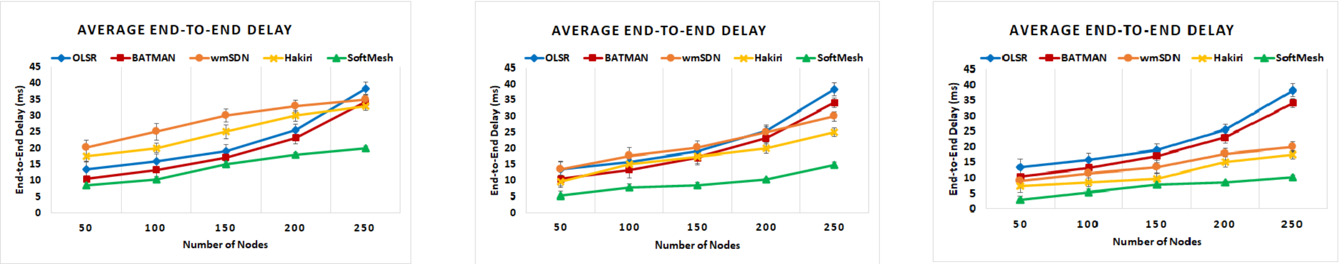


a) Scenario 01 (10% SDN hybrid nodes + 90% Legacy nodes)

b) Scenario 02 (25% SDN hybrid nodes + 75% Legacy nodes)

c) Scenario 03 (50% SDN hybrid nodes + 50% Legacy nodes)

FIGURE 12. Packet drop ratio.

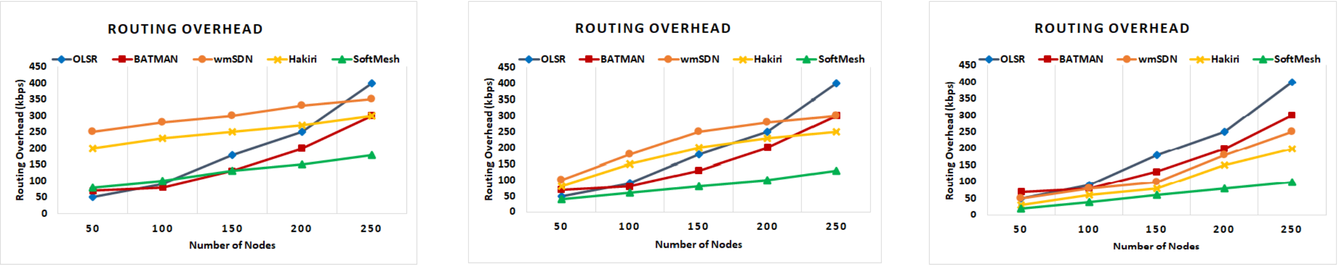


a) Scenario 01 (10%SDN hybrid nodes + 90%Legacy nodes)

b) Scenario 02 (25%SDN hybrid nodes + 75%Legacy nodes)

c) Scenario 03 (50%SDN hybrid nodes + 50%Legacy nodes)

FIGURE 13. **Average end-to-end delay.**



a) Scenario 01 (10%SDN hybrid nodes + 90%Legacy nodes)

b) Scenario 02 (25%SDN hybrid nodes + 75%Legacy nodes)

c) Scenario 03 (50%SDN hybrid nodes + 50%Legacy nodes)

FIGURE 14. **Routing overhead.**

collection delays, queuing delays, transmission, and propagation. Therefore, by assuming half of the RTT, we estimated the Round-Trip Time (RTT) and determined the one-way latency. Besides, we calculated the delay necessary to send a packet to the controller before it receives its router closure. The controller attempts to solve the problem of mobility, node, and connection failure by using the periodic broadcasting of route request messages. The results show that the end-to-end delay of legacy routing schemes, existing hybrid and Soft-Mesh architecture is higher in 50 and 100 node network topologies. Whereas for network topology of 150, 200, and 250 nodes, end-to-end delay is more in legacy routing schemes than in SDN networks. In legacy networks, each packet must be queried and forwarded after the routing table has been defined. The routing table scale increases with the growth of the network scale, resulting in the slow speed of querying and forwarding. In SDN, the data packet received by the node is first forwarded to the controller, then the flow entries are pushed to the node by the controller, after which the packets can be forwarded via the flow table query. In similar network topology, the OpenVswitch flow table scale is smaller than the router routing table scale, so when the network scale is high, the OpenVswitch forwarding speed will be higher.

D. ROUTING OVERHEAD

The global routing load incurred by routing protocols increases proportionally with the size of the network, as shown in Figure 14. We should also note that OLSR, while the best in terms of network throughput, is very costly in overhead among traditional routing protocols (OLSR and BATMAN). The centralized SDN operation removes the need for flooding used by the OLSR protocol, thus enabling better results even when the message exchange rate is high between the controller and the hosts. Therefore, using the SDN method and therefore having less overhead will minimize the effect of interference as well as convergence time.

Analysis of simulation results shows that centralized approach and out-of-band signaling of the Soft-Mesh solution enable WMNs to overcome the limitations caused by distributed routing. SDN routing is based on a centralized approach that establishes communication between nodes via the controller. When a node needs to connect to another node, the controller is asked to provide information about the path to that specific node, resulting in the implementation of the appropriate routing rules. In this respect, connection to the controller node plays a major role in establishing connectivity from one node to another. Although routing rules do not change or update periodically on wired networks due to the static nature of network nodes, resulting in no impact on the controller's node connectivity. However, communication between nodes, especially in wireless mesh networks, it is very much influenced by frequent node movement in wireless networks. The proposed Soft-Mesh routing architecture is more helpful for WMN compared to traditional BATMAN and OLSR routing protocols, and with other hybrid (SDN/IP) routing approaches. The subsequent methods aim to find the best route to the controller, resulting in a greater delay in establishing connections between the controller and the node, making it possible for the controller to remain unavailable for a longer period.

VII. CONCLUSION AND FUTURE WORK

This paper presents a robust routing architecture for hybrid SDN and wireless mesh networks. The main objective of this article is to combine SDN with WMN based on a hybrid topology and to examine routing problems and their effects while changing the SDN node architecture. Network topology is based on legacy nodes and SDN hybrid nodes consider the proposed hybrid routing architecture. The suggested solution is to hybridize SDN nodes and co-exist OLSR routing for IP-based forwarding with the OpenFlow protocol for SDN forwarding to achieve seamless interoperability between SDN and legacy nodes. It should be noted that our research does not seek to support the replacement of traditional routing approaches by SDN routing. The goal of this work is to investigate whether an architecture based on SDN can assist WMN routing and not arbitrarily equate legacy routing protocols (in-band signaling) and a centralized approach (out-of-band signaling). Considering legacy protocols, it is discussed how the SDN networking architecture can be used and to what degree the former can assist the latter. Moreover, Soft-Mesh architecture provides a cost-effective solution and seamless interoperability between legacy nodes and SDN nodes as compared to other hybrid routing architectures. Experiments are performed for the topology consisting of a varying number of SDN hybrid nodes and legacy nodes that are 50, 100, 150, 200, and 250 nodes respectively. Moreover, the proportion of SDN hybrid nodes and legacy nodes is also variable to create three different simulation scenarios including 10%, 25%, and 50% of SDN hybrid nodes whereas 90%, 75%, 50% of legacy nodes in the network topology. The size of topology has also been kept variable that is the size of 500m*500m for 50 and 100 nodes topology and 1000m*1000m for 150, 200, and 250 nodes topology. Simulations are carried out for OLSR and BATMAN routing schemes as these schemes are considered relatively more stable among all other traditional routing approaches for wireless mesh networks. However, for hybrid approaches, Soft-Mesh architecture has been compared with wmSDN and Hakiri. the proposed routing Soft-Mesh provides enhanced results in terms of various performance metrics including average UDP throughput, end-to-end delay, packet drop ratio, and routing overhead. Soft-Mesh gives 50% to 70% improved results for the incremental proportion of SDN hybrid nodes. Therefore, our findings indicate that the SDN approach will positively help the operations of the distributed routing protocol. As future work, we plan to validate the proposed hybrid routing architecture using some larger-scale testbed, and to analyze further changes SDN solution should introduce to the IP routing domain in WMNs at large. Moreover, we intend to apply SDN-based WMNs solutions to overcome the main challenges in the domain of IoT, highlighting their advantages, and exposing their weaknesses.

REFERENCES

[1] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Comput. Netw.*, vol. 72, pp. 74–98, Oct. 2014.

[2] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[3] F. D. O. Silva, J. H. D. S. Pereira, P. F. Rosa, and S. T. Kofuji, "Enabling future Internet architecture research and experimentation by using software defined networking," in *Proc. Eur. Workshop Softw. Defined Netw.*, Oct. 2012, pp. 73–78c.

[4] A. Drescher, "A survey of software-defined wireless networks," Dept. Comput. Sci. Eng., Washington Univ. St. Louis, St. Louis, MO, USA, Tech. Rep., 2014, pp. 1–15.

[5] J. F. Gonzalez Orrego and J. P. U. Duque, "Throughput and delay evaluation framework integrating SDN and IEEE 802.11s WMN," in *Proc. IEEE 9th Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2017, pp. 1–6.

[6] C. Chaudet and Y. Haddad, "Wireless software defined networks: Challenges and opportunities," in *Proc. IEEE Int. Conf. Microw., Commun., Antennas Electron. Syst. (COMCAS)*, Oct. 2013, pp. 1–5.

[7] A. Maleki, "An SDN perspective to mitigate the energy consumption of core networks-GÉANT2," in *Proc. Int. SEEDS Conf.*, Sep. 2017, pp. 1–13.

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

[9] S. S. A. Gilani, A. Qayyum, R. N. B. Rais, and M. Bano, "SDNMesh: An SDN based routing architecture for wireless mesh networks," *IEEE Access*, vol. 8, pp. 136769–136781, 2020.

[10] T. Bakhshi, "State of the art and recent research advances in software defined networking," *Wireless Commun. Mobile Comput.*, vol. 2017, pp. 1–35, Jan. 2017.

[11] H. Huang, P. Li, S. Guo, and W. Zhuang, "Software-defined wireless mesh networks: Architecture and traffic orchestration," *IEEE Netw.*, vol. 29, no. 4, pp. 24–30, Jul. 2015.

[12] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2004, pp. 114–128.

[13] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2713–2737, 4th Quart., 2016.

[14] M. L. Sichitiu, "Wireless mesh networks: Opportunities and challenges," in *Proceedings of World Wireless Congress*, vol. 2. Pittsburgh, PA, USA: CiteSeer, 2005.

[15] M. Eslami, O. Karimi, and T. Khodadadi, "A survey on wireless mesh networks: Architecture, specifications and challenges," in *Proc. IEEE 5th Control Syst. Graduate Res. Colloq.*, Aug. 2014, pp. 219–222.

[16] S. Y. Shahdad, A. Sabahath, and R. Parveez, "Architecture, issues and challenges of wireless mesh network," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 0557–0560.

[17] A. OlanrewajuAjayi, A. A. Adigun, and W. O. Ismaila, "A review of routing protocols for practical rural wireless mesh networks (WMNs)," *Int. J. Comput. Appl.*, vol. 114, no. 16, pp. 14–20, Mar. 2015.

[18] I. F. Akyildiz, "Research challenges for traffic engineering in software-defined networks," *IEEE Netw.*, vol. 30, no. 3, pp. 52–58, May 2016.

[19] P. Lin, J. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K.-C. Wang, and J. Bi, "Seamless interworking of SDN and IP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 475–476, Sep. 2013.

[20] V. Nascimento, M. Moraes, R. Gomes, B. Pinheiro, A. Abelem, V. C. M. Borges, K. V. Cardoso, and E. Cerqueira, "Filling the gap between software defined networking and wireless mesh networks," in *Proc. 10th Int. Conf. Netw. Service Manage. (CNSM) Workshop*, Nov. 2014, pp. 451–454.

[21] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in SDN/OSPF hybrid network," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 563–568.

[22] C. E. Rothenberg, A. L. L. A. N. Vidal, and M. Salvador, "Hybrid networking towards a software-defined era," in *Network Innovation Through OpenFlow and SDN: Principles and Design*. Boca Raton, FL, USA: CRC Press, 2014, pp. 153–198.

[23] M. Bano, S. S. A. Gilani, and A. Qayyum, "A comparative analysis of hybrid routing schemes for SDN based wireless mesh networks," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun.; IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1189–1194.

[24] Y. Peng, X. Gong, L. Guo, and D. Kong, "A survivability routing mechanism in SDN enabled wireless mesh networks: Design and evaluation," *China Commun.*, vol. 13, no. 7, pp. 32–38, Jul. 2016.

[25] M. Bano, "SDNHybridMesh: A hybrid routing architecture for SDN based wireless mesh networks," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2020, pp. 366–375.

[26] D. Levin, "Panopticon: Reaping the benefits of incremental SDN deployment in enterprise networks," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 333–345.

[27] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 70–75, Apr. 2014.

[28] Y. Peng, L. Guo, Q. Deng, Z. Ning, and L. Zhang, "A novel hybrid routing forwarding algorithm in SDN enabled wireless mesh networks," in *Proc. IEEE IEEE 17th Int. Conf. High Perform. Comput. Commun. 7th Int. Symp. Cyberspace Saf. Secur., IEEE 12th Int. Conf. Embedded Softw. Syst.*, Aug. 2015, pp. 1806–1811.

[29] J. Moy, *OSPF Protocol Analysis*, document RFC 1245, 1991.

[30] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[31] Y. Guo, Z. Wang, X. Yin, X. Shi, J. Wu, and H. Zhang, "Incremental deployment for traffic engineering in hybrid SDN network," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–8.

[32] M. Labraoui, M. M. Boc, and A. Fladenmuller, "Software defined networking-assisted routing in wireless mesh networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 377–382.

[33] J. Wang, "A software-defined network routing in a wireless multihop network," *J. Netw. Comput. Appl.*, vol. 85, pp. 76–83, May 2017.

[34] Y. Wei, X. Zhang, L. Xie, and S. Leng, "Energy-aware traffic engineering in hybrid SDN/IP backbone networks," *J. Commun. Netw.*, vol. 18, no. 4, pp. 559–566, Aug. 2016.

[35] L. He, X. Zhang, Z. Cheng, and Y. Jiang, "Design and implementation of SDN/IP hybrid space information network prototype," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, Jul. 2016, pp. 1–6.

[36] U. Ashraf, "Placing controllers in software-defined wireless mesh networks," in *Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Mar. 2018, pp. 1–4.

[37] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for wireless mesh networks," in *Proc. 20th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2011, pp. 1–6.

[38] D. Nguyen and P. Minet, "Analysis of MPR selection in the OLSR protocol," in *Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINAW)*, May 2007, pp. 887–892.

[39] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless mesh software defined networks (wmSDN)," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2013, pp. 89–95.

[40] S. Salsano, G. Siracusano, A. Detti, C. Pisa, P. L. Ventre, and N. Blefari-Melazzi, "Controller selection in a wireless mesh SDN under network partitioning and merging scenarios," 2014, *arXiv:1406.2470*. [Online]. Available: http://arxiv.org/abs/1406.2470

[41] H. Elzain and W. Yang, "QoS-aware topology discovery in decentralized software defined wireless mesh network (D-SDWMN) architecture," in *Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI)*, 2018, pp. 158–162.

[42] S. Salsano, P. L. Ventre, F. Lombardo, G. Siracusano, M. Gerola, E. Salvadori, M. Santuari, M. Campanella, and L. Prete, "Hybrid IP/SDN networking: Open implementation and experiment management tools," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 1, pp. 138–153, Mar. 2016.

[43] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," *IEEE Netw.*, vol. 28, no. 2, pp. 42–48, Mar. 2014.

[44] A. Hakiri, A. Gokhale, and P. Patil, "A software-defined wireless networking for efficient communication in smart cities," Distrib. Object Comput. (DOC) Group, Vanderbilt Univ., Nashville, TN, USA, Tech. Rep., 2017.

[45] A. Hakiri, A. Gokhale, and P. Berthou, "Software-defined wireless mesh networking for reliable and real-time smart city cyber physical applications," in *Proc. 27th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2019, pp. 165–175.

[46] H. Elzain and Y. Wu, "Software defined wireless mesh network flat distribution control plane," *Future Internet*, vol. 11, no. 8, p. 166, Jul. 2019.

[47] Shastry, Nithin, and TG Keerthan Kumar, "Enhancing the performance of software-defined wireless mesh network," in *Proc. Int. Conf. Commun., Comput. Electron. Syst* Singapore: Springer, 2020, pp. 1–14.

[48] E. Kuznetsova, Y. Avakyan, and R. Kirichek, "Applying the concept of software-defined networking in wireless mesh network," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, Cham, 2020, pp. 28–38.

[49] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.

[50] M. Malboubi, L. Wang, C.-N. Chuah, and P. Sharma, "Intelligent SDN based traffic (de)Aggregation and measurement paradigm (iSTAMP)," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 934–942.

[51] M. Moshref, M. Yu, and R. Govindan, "Resource/accuracy tradeoffs in software-defined measurement," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 73–78.

[52] Y. Zhang, "An adaptive flow counting method for anomaly detection in SDN," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2013, pp. 25–30.

[53] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," in *Proc. Int. Conf. Passive Act. Netw. Meas.* Berlin, Germany: Springer, 2010, pp. 201–210.

[54] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A low cost network monitoring framework for software defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[55] C. Yu, "Flowsense: Monitoring network utilization with zero measurement cost," in *Proc. Int. Conf. Passive Active Netw. Meas.*, Berlin, Germany: Springer, 2013, pp. 31–41.

[56] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "DREAM: Dynamic resource allocation for software-defined measurement," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 419–430, 2015.

[57] M. Hartung and M. Körner, "SOFTmon—Traffic monitoring for SDN," *Procedia Comput. Sci.*, vol. 110, pp. 516–523, Jan. 2017.

[58] K. Bao, J. D. Matyjas, F. Hu, and S. Kumar, "Intelligent software-defined mesh networks with link-failure adaptive traffic balancing," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 2, pp. 266–276, Jun. 2018.

[59] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, and M. F. Magalhães, "QuagFlow: Partnering quagga with OpenFlow," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 441–442, Aug. 2010.

[60] W. Wang, C.-H. Wang, and T. Javidi, "Reliable shortest path routing with applications to wireless software-defined networking," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[61] D. Sajjadi, Z. Zheng, R. Ruby, and J. Pan, "Randomized single-path flow routing on SDN-aware Wi-Fi mesh networks," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2018, pp. 184–192.

[62] J. Kim, S. Lee, and S. Lee, "Mesh network convergence management system using software-defined network," *Mobile Inf. Syst.*, vol. 2018, Jan. 2018, Art. no. 7157948.

[63] P. Bellavista, A. Dolci, C. Giannelli, and D. D. P. Montenero, "SDN-based traffic management middleware for spontaneous WMNs," *J. Netw. Syst. Manage.*, vol. 28, no. 4, pp. 1575–1609, Oct. 2020.

[64] Y. Wang, T.-Y.-C. Tai, R. Wang, S. Gobriel, J. Tseng, and J. Tsai, "Optimizing open vSwitch to support millions of flows," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–7.

[65] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, *Optimized Link State Routing Protocol (OLSR)*, document HAL ID: inria-00471712, HAL-Inria, 2003.

[66] N. Makariye, "Towards shortest path computation using Dijkstra algorithm," in *Proc. Int. Conf. IoT Appl. (ICIOT)*, May 2017, pp. 1–3.

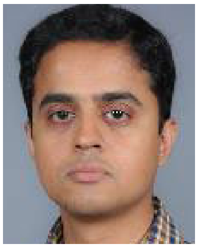
[67] Y. Chai and X.-J. Zeng, "The development of green wireless mesh network: A survey," *J. Smart Environ. Green Comput.*, vol. 30, pp. 47–59, Mar. 2021.

[68] A. Eftimie and E. Borcoci, "SDN controller implementation using OpenDaylight: Experiments," in *Proc. 13th Int. Conf. Commun. (COMM)*, Jun. 2020, pp. 477–481.

[69] D. Johnson, N. Ntlatlapa, and C. Aichele, "Simple pragmatic approach to mesh routing using BATMAN," Council Sci. Ind. Res., Pretoria, South Africa, Tech. Rep., 2008.

MUKHTIAR BANO (Member, IEEE) received the M.S. degree in computer engineering from the Center for Advanced Studies and Engineering (CASE) Islamabad, UET at Taxila, Pakistan, in 2011. She is currently pursuing the Ph.D. degree in computer engineering with the Capital University of Science and Technology, Islamabad. She has worked as an IT Officer at Pakistan Air Force, Pakistan, from 2001 to 2005. She is also working as an Assistant Professor with the Department of Software Engineering, Fatima Jinnah Women University, Rawalpindi, Pakistan. Her research interests include software-defined networks, wireless mesh networks, network monitoring, and routing in wireless networks.

AMIR QAYYUM (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the University of Engineering & Technology at Lahore, Lahore, in 1991, and the M.S. and Ph.D. degrees from Paris-Sud University, France, in 1996 and 2000, respectively. He completed his research work at INRIA, Rocquencourt, France. He was associated with INRIA Rocquencourt, France, for research on MANETs, and coauthored optimized link state routing (OLSR) protocol RFC at the Internet engineering task force (IETF). He is currently working as a Professor with the Capital University of Science and Technology (CUST), Islamabad, Pakistan. He is also the Head of the Center of Research in Networks and Telecom (CoReNeT). He has led several national and international and funded research projects in the domain of wired and wireless networks. He has several publications in international conferences and journals. His research interests include wireless and mobile computer networks, multimedia networks, software-defined networking, vehicular and mobile *ad hoc* networks, and next-generation protocols. He is the Chair of the IEEE Islamabad Section.



RAO NAVEED BIN RAIS (Member, IEEE) received the B.E. degree in computer systems from the National University of Sciences & Technology, Pakistan, in 2002, and the M.S. and Ph.D. degrees from the University of Nice Sophia Antipolis, France, in 2007 and 2011, respectively. He completed his research work at INRIA, Sophia Antipolis, France. He has worked with COMSATS University and the Capital University of Science & Technology, Islamabad, Pakistan. He carries industrial research and development, academic, and research experience for several years. He is currently working as an Associate Professor with Ajman University, Ajman, United Arab Emirates. He has several publications in renowned international conferences and journals. His research interests include next-generation computer networks, protocol design and development, software-defined networking, information-centric networking, and opportunistic networking.

SYED SHERJEEL A. GILANI received the M.S. degree in computer science from Riphah International University, in 2011. He is currently pursuing the Ph.D. degree in computer engineering with the Capital University of Science and Technology, Islamabad. From 2009 to 2013, he has worked as the Manager (IT) with Foundation University, Pakistan, and a Lecturer with the PAF College JC, Chaklala, Pakistan, from 2001 to 2009. He is also working as a Senior Lecturer with the Faculty of Computing, Riphah International University, Islamabad, Pakistan. His research interests include software-defined networks, wireless mesh networks, the Internet of Things, and routing in wireless networks.

• • •