IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

A Survey of Hybrid Software Defined Networks: Challenges, Opportunities, and Future Directions

TIAGO AFONSO MARQUES[1]
tamarques@ua.pt, 98459

ABSTRACT The evolution of network management through Software Defined Networks (SDNs) has ushered in a new era of flexibility and scalability, primarily facilitated by centralised controllers. In this paradigm, SDN controllers employ OpenFlow Discovery Protocol (OFDP) to establish a comprehensive view of the network topology. While crucial for effective routing and application deployment, the OFDP in a Hybrid Software Defined Network (hSDN) network environment introduces challenges such as repetitive operations, diminishing overall effectiveness, efficiency, and scalability. Moreover, its inherent limitations restrict its ability to detect diverse network devices beyond SDN components. This paper sheds light on the challenges posed by the OFDP in a hybrid SDN environment and explores related works aiming to address this multifaceted problem. The focus is on the necessity for an accurate global view of the network topology, crucial for optimizing routing, load balancing, and supporting mobility-based applications. By delving into these challenges and examining potential solutions proposed in existing literature, this work contributes to the ongoing discourse on advancing topology discovery mechanisms for the evolving landscape of hybrid SDN networks.

I. INTRODUCTION

As networks have grown larger and more complex, the demand for increasingly intricate services, network policies, and rules has surged. The implementation and management, while navigating through vendor-specific configurations, of these evolving needs within traditional networking have become unmanageable and a cumbersome process. The requirement to configure each device individually further exacerbates the inherent complexity of addressing these evolving demands. Furthermore, the limitations of traditional protocols contribute to the growing complexity. These protocols often lack the flexibility required to seamlessly adapt to the dynamic demands of modern networks. When alterations are made to one aspect of the traditional network, the interdependency among various policies, rules, and configurations can lead to unintended consequences. Since the control plane is decentralized, modifications in one device may inadvertently impact other elements within the network, creating a ripple effect of potential issues. This lack of centralized control makes troubleshooting and ensuring consistent policies across the entire network challenging. Given the challenges posed by the rigid nature of traditional protocols and the decentralized control plane structure, the need for a more agile and centralized approach becomes evident.

In response to this imperative, the paradigm of SDNs emerges as a beacon of innovation. This new architecture not only addresses the hurdles posed by traditional protocols but also introduces a revolutionary shift in network management. Beyond satisfying the pressing need for agility and centralized control, SDN brings additional features and capabilities to the forefront. A significant advancement lies in the flexibility and programmability of SDNs. SDNs exhibit the capability to adapt on the fly to changes in network topology, whether it involves the addition or removal of devices. This dynamic adaptability extends beyond topology changes, harnessing the programmable nature of SDNs to enable smooth and efficient adjustments. These adjustments, encompass not only shifts in the network topology, but also the implementation of changes in network policies based on various variables.

In navigating the complexities of modern network environments, it's essential to recognize the unique strengths that both traditional and SDNs paradigms bring to the table. Traditional networking, with its established protocols and decentralized control, forms the backbone of many existing infrastructures, providing reliability and stability. On the other hand, SDNs introduces a revolutionary approach, emphasizing centralized control and dynamic adaptability. Its flexibility and programmability enable networks to swiftly respond to changes in topology and adjust policies efficiently.

However, as networks expand and demands become more intricate, the limitations of each approach become apparent. Traditional networks grapple with the challenge of managing

complex configurations across diverse devices, while SDNs faces the need to integrate seamlessly with existing infrastructures. Recognizing these strengths and limitations, a natural evolution emerges, the hSDN.

In the realm of hSDNs, the integration of diverse network elements goes beyond traditional and SDN switches. Leveraging the inherent flexibility and programmability of SDNs, this hybrid model extends its reach to include unconventional yet highly relevant components, such as mobile devices. The programmable nature of SDNs allows for the dynamic inclusion of mobile devices like On-Board Units (OBUs) into the network infrastructure. These devices, with their ability to collect and transmit data in real-time, can contribute to the network's intelligence, offering valuable insights and enhancing the overall efficiency of the network. This dynamic integration enables the network to adapt on-the-fly to the mobility patterns and changing locations of these mobile nodes. Furthermore, within the hSDN framework, a noteworthy inclusion is the integration of Vehicular Ad-Hoc Networks (VANETs). These networks, formed by vehicles equipped with communication capabilities, contribute to the dynamic and evolving nature of the overall network. VANETs within the hybrid model bring about a new dimension, allowing vehicles to communicate with each other and with fixed infrastructure elements. This allows for the deployment of various Intelligent Transportation Systems (ITS) applications within the hSDNs.

In the hSDN paradigm, diverse node types, including mobile nodes (e.g., OBUs) and infrastructure (e.g., legacy and SDN switches, Road Side Units (RSUs)), demand support for various communication technologies. Mobile nodes benefit from 5G and cellular networks, providing high-speed, low-latency connectivity for real-time applications. Immobile components leverage fiber optic communication, ensuring reliable, high-bandwidth data exchange. This strategic integration of communication technologies underscores hSDN's adaptability, creating a cohesive network that efficiently addresses the diverse needs of its components.

II. SOFTWARE DEFINED NETWORK

On traditional networks the process of configuring, managing and operating said network makes it a lengthy and complex procedure. The people responsible for this task must deal with vendor-specific configurations to implement complex network topologies, policies and rules, as well as, configure each device individually. Another problem surrounding traditional networks is the lack of a standard when it comes to configuring said network devices and managing them. The devices are closed and work based on proprietary software, which creates even more additional difficulties. Considering all the difficulties outlined earlier, one can infer that the task is demanding, requires a significant amount of time, and is susceptible to mistakes.

Enter SDNs, a new approach that seeks to revolutionize the methodologies employed in the configuration, management, and operation of the network. Unlike traditional networks, where manual, device-specific configurations dominate, SDN brings a paradigm shift by centralizing network control trough software. In this new environment, the control plane, responsible for the flow of data trough the network, is decoupled from the data plane, which handles the forwarding of data packets using a set of rules specified by the control plane. The segregation of the control and data plane facilitates the establishment of a central logical controller wherein all network policies and rules are maintained and subsequently administered. This configuration not only centralizes the governance of the network but also serves to streamline the intricacies associated with network management [1].

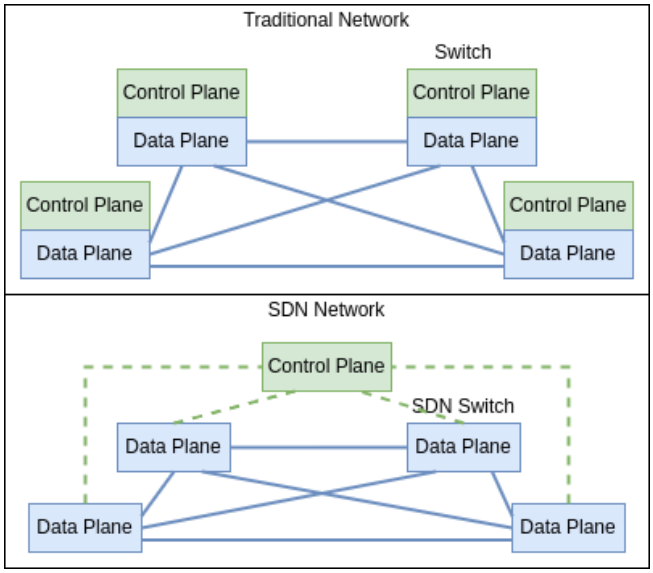

FIGURE 1. Traditional Network versus SDN (adapted from: [2])

SDNs outperforms traditional networks with centralized control for enhanced management, automation reducing errors, and dynamic adaptability to swiftly address changing needs. Its flexibility allows easy network adjustments, while vendor-agnostic abstraction simplifies device management. SDNs also offers cost efficiency trough minimized manual interventions and optimized resources utilization [3]:

A. ARCHITECTURE

At its fundamental design, the architecture of SDNs manifests a crucial segregation between the control plane, entrusted with decision-making regarding data routing, and the data plane, tasked with the actual transmission of data packets. This pivotal architectural paradigm shift facilitates the relocation of the control plane to a centralized location. Within this centralized position, a cohesive logic unit emerges, affording network engineers and administrators a comprehensive overview of the entire network, thereby enabling more efficient management.

The architecture comprises three distinct layers, each serving a specific purpose in this innovative networking model [3]:

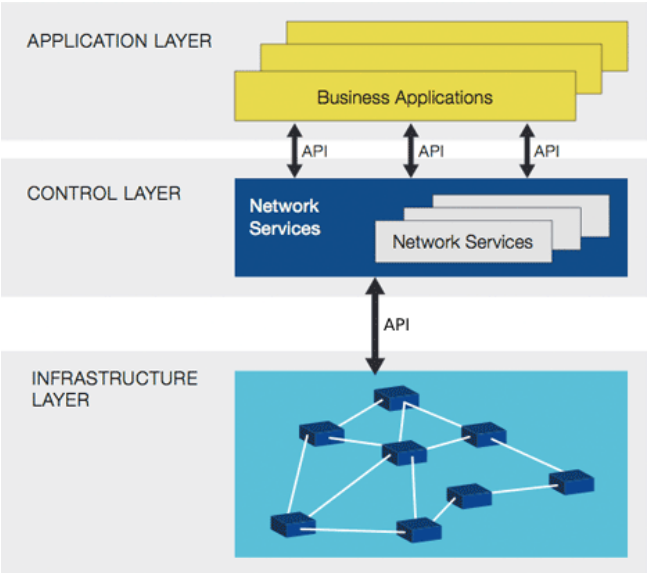

FIGURE 2. SDN Architecture (adapted from [4])

1) Application Layer

Serving as the uppermost layer within the SDN model, the application layer functions as the pinnacle of hierarchical organization. It accommodates software applications, which, in turn, engage in interactions with the SDN controller through northbound Application Programming Interfaces (APIs). In this capacity, applications have privileged access to a comprehensive and unified perspective of the entire network infrastructure. Consequently, they possess the capability to institute new policies and enact strategic implementations, leveraging their global visibility for effective orchestration and management.

2) Control Layer

Situated as the intermediary layer, the control layer embodies a pivotal role in the hierarchical architecture. This layer encompasses the centralized SDN controller, which serves as the orchestrator of network operations. Tasked with processing high-level instructions emanating from applications in the application layer, the controller employs southbound Application Programming Interfaces (APIs) to communicate directives to network devices residing in the data layer. In its essence, the control layer operates as the command center, translating abstract policies and decisions into actionable configurations for network devices. This centralized control mechanism imparts a cohesive and streamlined approach to network management, ensuring uniformity and consistency across the entire infrastructure. Moreover, the control layer enables dynamic adaptability by facilitating real-time adjustments to network configurations. Through its centralized intelligence, the SDN controller not only responds to changing conditions but also provides a comprehensive vantage point for administrators, affording them enhanced visibility and control over the network. This layered architecture, with the control layer at its core, forms the backbone of SDN, fostering programmability, centralization, and efficient orchestration of network resources.

3) Data Layer

Constituting the foundational tier in the SDN model, the data layer is integral to the network infrastructure, encompassing network devices such as switches and routers. Operating in a streamlined capacity within the SDN architecture, these devices assume the responsibility of forwarding data packets based on instructions emanating from the SDN controller, which resides in the control layer. The data layer serves as the workhorse, executing the directives communicated through southbound APIs from the control layer. These southbound interfaces, including notable protocols like OpenFlow (OF), facilitate seamless communication between the controller and the network devices. Through this interface, the control layer transmits specific configurations and policies to the data layer, influencing the behavior of individual devices. In essence, the southbound interface acts as the conduit through which the centralized intelligence of the control layer is disseminated to the distributed to the network infrastructure. The data layer, with its network devices acting based on instructions from the control layer, ensures the precise and coherent forwarding of data packets throughout the network.

This architectural configuration, comprising the collaborative interplay of the application, control, and data layers, describes the nature of SDNs in modern networking paradigms. It introduces a dynamic and responsive approach to network management, optimizing resource utilization and enhancing overall operational efficiency.

B. CONTROLLER

The SDN controller serves as the fundamental element in the SDN architecture, entrusted with the vital role of orchestrating communication between network applications and the data plane. This interaction occurs through the northbound and southbound interfaces, respectively. Referred to as the Network Operating System (NOS), the SDN controller acts as a facilitator for network control logic, offering the application layer a high-level, abstracted perspective of the entire network.

By adopting the term Network Operating System, the controller abstracts the intricacies of implementation, presenting a coherent and comprehensible view for network applications. This abstracted perspective furnishes sufficient information to formulate policies while concealing the operational details. Despite the numerous advantages that SDN introduces in comparison to traditional networking, there linger concerns about its scalability and reliability when deployed in extensive network environments. Ongoing scrutiny is directed towards the seamless integration and dependability of SDN in large-scale network scenarios [2].

Key features of the SDN controller include:

- **Centralized Control:** The SDN controller serves as a centralized logical point, responsible for the decision-making processes. This centralized control allows for a unified and coherent management approach across the entirety of the network.
- **Communication Interfaces:** Equipped with northbound APIs, used for receiving high-level instructions and policies from applications, and southbound APIs, that facilitate communication with network devices in the data layer, the SDN controller establishes communication links with both the application layer above and the data layer below.
- **Dynamic Adaptability:** The controller is designed to be dynamically adaptable, responding in real-time to changing network conditions and requirements. This adaptability is crucial for the efficient management and optimization of network resources.
- **Standardized Protocols:** SDN controllers typically employ standardized protocols, such as OF, to communicate with network devices. OF, in particular, facilitates the exchange of information between the controller and switches, enabling granular control over packet forwarding.
- **Global Network View:** As a consequence of its centralized position, the SDN controller provides, to network engineers and administrators, a comprehensive and global view of the network.
- **Policy Translation:** One of the core functions of the SDN controller is to interpret and translate high-level policies received from applications into specific configurations and directives comprehensible to network devices. This translation ensures the seamless execution of policies throughout the network.

At its core, the SDN controller plays a pivotal role in SDN architecture, embodying centralized intelligence that facilitates the seamless integration of high-level policies and decisions throughout the distributed network. Serving as the orchestrator of network operations, the SDN controller interprets and translates abstract policies from the application layer into actionable configurations for network devices in the data layer.

C. SOUTHBOUND PROTOCOLS/APIS

SDN controllers have the flexibility to employ various southbound protocols and APIs in order to achieve a unified network management, Figure 3. In the context of this article, particular emphasis will be placed on discussing the used protocols.

1) OpenFlow

SDN has revolutionized the field of networking, introducing dynamic control and programmability into network management. Among the various protocol standards, OF has emerged as a prominent and widely adopted solution, offering a versatile approach to implementing SDN concepts.

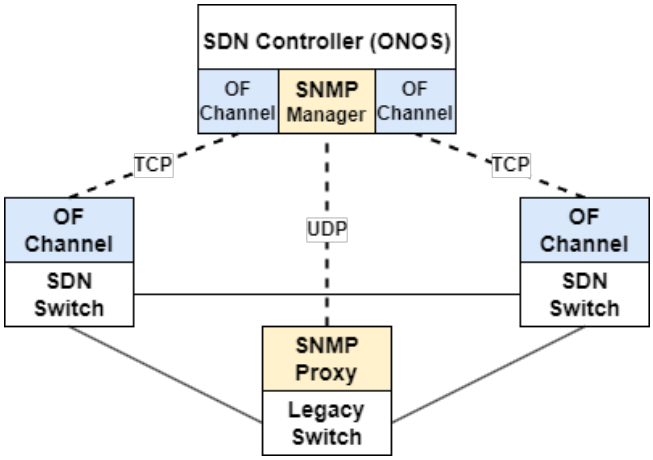

FIGURE 3. Unified Network Management Example

OF serves as the standard communications interface between the control and data layers within SDN architectures. By enabling direct access and manipulation of the forwarding plane in network devices, including switches and routers, whether physical or virtual. OF breaks away from the monolithic, closed, and mainframe-like nature of legacy networking devices, addressing the absence of an open interface to the forwarding plane.

Unlike other protocols, OF facilitates the decentralization of network control from switches to logically centralized control software.

Implemented on both ends of the interface between network infrastructure devices and SDN control software, it employs flows to identify network traffic based on pre-defined match rules. These rules can be programmed dynamically or statically by the SDN control software, allowing for network engineers and administrators to dictate how traffic flows through network devices based on factors such as usage patterns, applications, and cloud resources.

One of OF's distinctive features is its capability to program the network on a per-flow basis, providing exceptionally granular control. This level of precision allows the network to adapt swiftly to real-time changes at the application, user, and session levels. In contrast, conventional IP-based routing lacks this fine-grained control, as all flows between two endpoints are typically constrained to follow the same path through the network, regardless of their distinct requirements.

As a key enabler for SDN, OF offers a standardized protocol that facilitates the direct manipulation of the forwarding plane in network devices. Initially designed for Ethernet-based networks, OF switching has extended its applicability to include terminal nodes, enabling direct SDN interactions between the network and mobile nodes. This extension optimizes mobility processes, allowing mobile nodes to efficiently convey information about connectivity targets and conditions through a unified control protocol. This integration of SDN mechanisms with mobile nodes grants the controller the capability to manage data flows seamlessly up to the

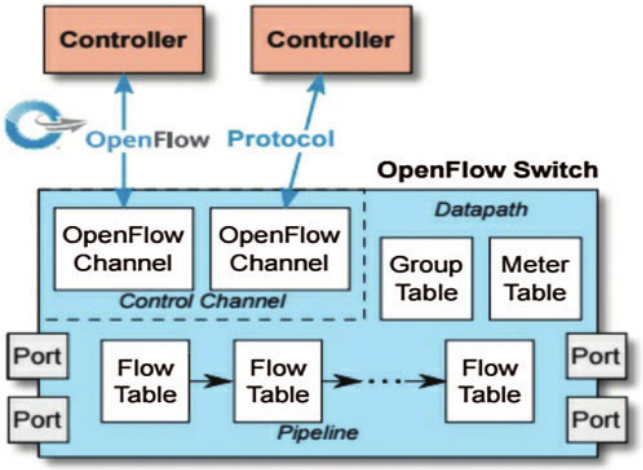

FIGURE 4. OF Switch Components [5]

terminal node.

OF Switch - The OF Logical Switch in SDN consists of flow tables, a group table, and OF channels linking it to an external controller. This switch communicates with the controller through OF channels, enabling the controller to manage the switch via the OF switch protocol.

Flow tables within the switch are responsible for packet lookup and forwarding. The controller, using the OF switch protocol, can add, update, or delete flow entries in these tables, defining how packets are processed based on specific criteria. When a packet arrives, it undergoes matching against flow entries in the flow tables. These entries specify actions for matching packets, such as forwarding, modification, or group processing. Packets can be forwarded to different ports, including physical, logical, or reserved ports with generic forwarding actions. Group processing, facilitated by the group table, allows additional actions for packets, such as multipath routing or link aggregation.

Switch designers possess the freedom to structure internal components with flexibility. OF based SDN seamlessly blend into existing network infrastructures, accommodating both OF-based and traditional forwarding approaches. This versatility facilitates a gradual implementation of OF technologies in a wide array of network environments [5].

2) Simple Network Management Protocol (SNMP)

SNMP operates within the Client/Server model, featuring a managing process (Manager) and a proxy process (SNMP proxy). The Manager is a software module situated in a network management system, providing a user-friendly interface for configuring and managing network operations. Conversely, the SNMP proxy is pre-installed in managed network devices, responsible for local information management and communicating with the managing process. SNMP utilizes User Datagram Protocol (UDP) as its transport protocol, and messages are exchanged between the managing process and proxy process. This protocol is commonly used for monitoring and controlling network devices, providing valuable information for network management tasks [6].

In the context of SDNs, many legacy devices support SNMP, making it a valuable tool. SDN controllers, acting as the manager, can use this protocol to manage and interact with traditional network elements, allowing for seamless integration and centralized control, even when devices do not natively support SDN protocols. This capability simplifies the management of diverse network infrastructures within the SDN framework.

D. IMPLANTATION MODELS

SDN implementations involve a controller, switches, and a communication protocol. These elements, separated in SDN, efficiently forward and route data packets, providing organizations with flexible implementation options. The following models provide different transitions and deployment options [7]:

- **Open SDN:** In this basic SDN type, the controller utilizes a standard southbound protocol, like OF, to communicate with network switches. The controller receives information from applications, converts it into flow entries, and transmits them via OF to switches. This centralized control allows organizations to oversee data movement across all switches, whether virtual or physical, specifically designed for SDN-enabled devices only.
- **SDN via APIs:** This type employs southbound APIs to monitor and control network traffic in and out of switches. Unlike OF-based SDN, it allows switches to use traditional methods like SNMP or Network Configuration (NETCONF), as well as modern approaches like REST APIs. Specific control points for each device enable remote operation by the controller through APIs. This approach is compatible with traditional switches, simplifies orchestration software development, and promotes greater openness with reduced reliance on proprietary software and devices.
- **Overlay Model:** This model involves creating a virtual network on top of existing infrastructure, with various channels facilitating data flow to different data centers. Physical devices remain unchanged, and a hypervisor serves as the interface between them and the virtualized network. Each network channel is allocated a set bandwidth and specific devices. Only edge devices connect physically to the virtualized network. Data packets sent to an edge device, or Virtual Tunnel Endpoint (VTEP), are repackaged by the hypervisor and sent to the destination VTEP across the network, as directed by the controller.
- **Hybrid SDN:** These various implementations are not mutually exclusive; instead, they provide organizations with the flexibility to overlap and coexist based on specific needs. This approach accommodates both legacy and SDN devices within the network, allowing for a gradual transition. The versatile nature of SDNs enables

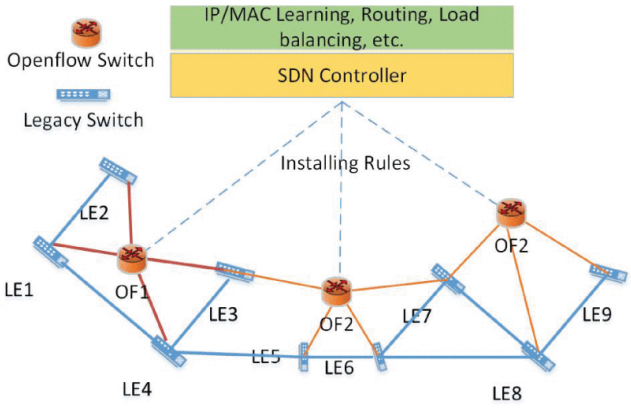

FIGURE 5. Example of hSDN [8]

the integration of different deployment models, resulting in a customized solution that caters to the unique requirements of different network segments. Although this approach offers flexibility aligned with organizational needs, it does introduce complexity, requiring skilled personnel for effective management and issue troubleshooting.

III. HYBRID SOFTWARE DEFINED NETWORKS

In a hSDN environment, both centralized and decentralized paradigms coexist, enabling a gradual transition from traditional networking to a more software-defined and dynamic infrastructure. hSDN represents an evolutionary step in networking architecture, seamlessly blending the robust foundations of traditional networking with the agility and programmability offered by SDNs. In this dynamic environment, the convergence of both centralized and decentralized control paradigms allows organizations to navigate the complexities of network transformation with a balanced and gradual approach.

Within the hSDN ecosystem, a diverse array of network devices play pivotal roles in shaping the hybrid network landscape. Legacy devices, representing traditional networking, stand alongside SDN devices that span both wired and wireless environments. The inclusion of dynamic or moving nodes further enriches the network fabric, adding a layer of adaptability to the overall architecture. Legacy devices, with their distributed algorithms, persist in providing a familiar foundation for certain network functions. Meanwhile, SDN devices bring programmability and adaptability to the forefront, allowing for dynamic adjustments based on the global network view provided by the SDN controller.

In this intricate mixture of diverse network devices, the hSDN environment thrives on the ability to accommodate the mobility and dynamism introduced by dynamic nodes. Whether it's adapting to changes in network topology or addressing the challenges posed by mobile devices, the hybrid SDN architecture showcases its resilience and adaptability.

A. ADVANTAGES AND DISADVANTAGES

Among the advantages of hSDN, the following are the most relevant [8]:

- **Cost Reduction:** provide a cost-effective approach to network deployment by allowing organizations to integrate SDN devices gradually alongside existing legacy infrastructure. This incremental strategy mitigates the financial burden associated with a complete network overhaul.
- **Partial Adoption of SDN Benefits:** Organizations can leverage select benefits of the SDN paradigm without committing to a full transition. hSDNs enable the strategic deployment of SDN devices in specific network segments, allowing for the optimization of resources and control where needed.
- **Detailed Control:** The selective implementation of SDNs in specific portions of the network enables detailed control over data traffic flows. hSDNs cater to the diverse needs of different network segments, enhancing overall efficiency and responsiveness.
- **Leveraging Traditional Routing Protocols:** Traditional routing protocols, effective for specific network tasks, can be employed in conjunction with SDNs in hybrid networks. This cooperative approach optimizes the use of both legacy and SDN technologies, improving overall network functionality.

There are different implementation models, therefore different models have different drawbacks, but in general the disadvantages are as follow [9]:

- **Higher Control Plane Complexity:** Interactions between multiple control planes may compromise the safety of network update procedures, leading to potential anomalies. Control plane conflicts can trigger forwarding inconsistencies, resulting in issues such as forwarding loops and traffic black holes. Establishing communication between the SDN controller and legacy switches is challenging, with various alternatives like middleware, protocol translation, and software upgrades presenting different trade-offs.
- **Increased Data Plane Complexity:** Implementing heterogeneity with a middleware layer to translate legacy protocols introduces added complexity in the data plane. This adaptation layer, while addressing heterogeneity, also increases latency and processing time. Introducing a middleware layer necessitates addressing security issues, ensuring replication for failure guarantees, and allocating extra processing power. Manual reconfiguration of legacy devices may lead to inappropriate or error-prone deployment, resembling challenges encountered in traditional legacy networks.
- **Scalability:** Controller scalability becomes a concern in hSDNs, as the controller's ability to control a limited number of devices is constrained, especially considering the overhead of interoperability. Incremental deploy-

ment of SDN devices may potentially lead to increased latency.

IV. TOPOLOGY DISCOVERY IN SDN

Since the SDN controller has no need to discover the SDN devices, because it is assumed that the devices will try to connect to the controller, the topology discovery mechanism is only concerned with the links between the network devices. The link discovery mechanism must be able to detect the links between the network devices, as well as, the ports that are connected to each link. Most of today's SDN controllers rely on a combination of OF Discovery Protocol (OFDP) and Link Layer Discovery Protocol (Link Layer Discovery Protocol (LLDP)) to discover the network topology.

A. OPEN SDN

The LLDP is typically implemented by ethernet switches, and is used to advertise the identity and capabilities of the device to it's neighbors. These packets are sent periodically, via each port of the switch, to a special multicast address to single hop neighbors and are not forwarded by the switches. The LLDP packets frame contains an LLDP Data Unit (Link Layer Discovery Protocol Data Unit (LLDPDU)), which contains a number of Type-Length-Values (TLVs) that contain information about the device, such as the device's name, port number, port description, etc. Switches, upon receiving LLDP packets, can extract the information from the LLDPDU and use it to update the device's Management Information Database (MIB) [10].

The OFDP is an extension of the OF protocol, which is used by the SDN controller to discover the network devices. It leverages the LLDP packets, but operates in a different manner. A standalone OF switch cannot send or process LLDP packets. The process of sending and processing said packets is handled by the SDN controller itself. The procedure works as follows [11]:

1) Firstly, the SDN controller creates an LLDP packet for each port of the switch. Each of these packets contains additional information, such as the switch's deviceID, port number and more. Then, the SDN controller sends these packets to the switch, trought OF Packet-Out message via the OF channel.
2) The switch upon receiving the LLDPs packets, from the controller, forwards them to the neighboring switches, trough the specified ports.
3) The neighboring switches receive the LLDPs packets and forwards them to the SDN controller via Packet-In messages via the OF channel. This is possible due to the fact that OF switches have a pre-installed rule in their flow table that states that LLDPs packet received from every port, with the exception of the controller port, must be forwarded to the controller. Lastly, the controller receives the LLDPs packets and extracts the information from the LLDPDU, updating the network topology database.

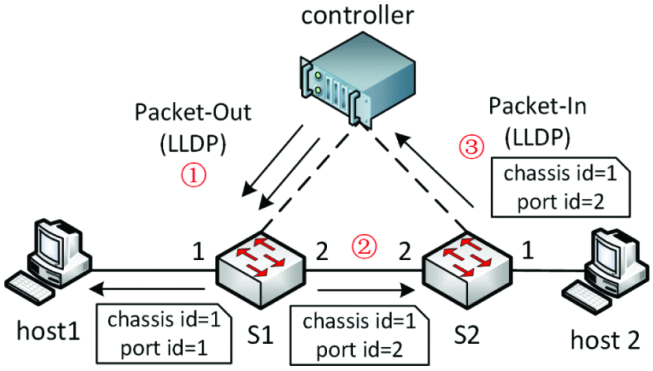

FIGURE 6. OFDP link discovery process [12]

B. HSDN

OFDP, originally designed to uncover one-hop links between SDN switches, falls short when it comes to identifying indirect multi-hop SDN links. To address this limitation, Broadcast Domain Discovery Protocol (BDDP) packets were introduced. BDDP packets, which carry the same content as the LLDP ones with the exception of the Ethernet type change, aim to enhance the discovery of more complex network paths.

Unlike LLDP packets, BDDP packets, specifically designed for hybrid SDN environments, are generally not processed or dropped by legacy switches. This distinction ensures that BDDP packets can effectively traverse the network, even when encountering legacy switches within the same broadcast domain. While BDDPs significantly improves the discovery of intricate network paths involving multiple hops, it's crucial to acknowledge that this packet is tailored to uncovering SDN links and is not equipped to discover legacy devices [13].

V. CHALLENGES

Despite its utility, the employed link discovery protocol, which uses both LLDP and BDDP packets, exhibit limitations, particularly when applied to dynamic and large-scale hybrid SDN networks.

The fisrt constraint is associated with the repetitive nature of link discovery, wherein controllers routinely interrogate the status of each link. This involves the periodic transmission of LLDP/BDDP packets to SDN switches, and corresponding active ports, within the network. As the network expands, the sheer volume of messages exchanged for link discovery and maintenance significantly rises. This increased traffic poses a potential risk of network congestion, imposing a substantial overhead on the controller's workload. Consequently, the extensive demands of this process can lead to operational challenges, impacting application-level network services, emphasizing the necessity to address the limitations of existing link discovery protocols [13].

To address the issue, the work in [12], proposes an extended version of OFDP, referred as OFDPx. It introduces a two-stage approach to link discovery, consisting of the initial stage and the update stage. In the first stage, the network view in the

SDN controller is established using methods such as OFDP. The update stage's primary purpose is to detect topology changes, particularly link failures, to update the network view promptly. It uses the discovered network links and divides them into multiple paths, each detected using a single probe packet, minimizing the number of messages exchanged and optimizing resource consumption.

Another approach is taken in [14], leaving the OFDP employed by OF, and proposing a new protocol called Enhanced Topology Discovery Protocol (eTDP). This protocol employs a software agent (eTDP client) in each network device to perform local discovery tasks and communicate relevant information to SDN controllers. This approach allows local SDN switches to perform certain tasks without requiring intervention from the central SDN controller. Consequently optimizing the efficiency of the topology discovery process while still maintaining a global view through the SDN controller.

Similarly in [15] new protocol was also proposed. In the Lightweight Automatic Discovery Protocol (LADP), the controller initiates network exploration by transmitting a single probe frame to a randomly selected root switch. The root switch then disseminates the LADP frame to neighboring switches, which, in turn, relay it to their respective neighbors while concurrently forwarding it to the SDN controller. By analyzing all received LADP frames, the SDN controller gains comprehensive information about all interconnected links within the network.

The second constraint pertains to the identification of diverse links within the network. This encompasses the incapacity to recognize unicast and broadcast legacy links, along with inter-domain links linking distinct SDN domains, each overseen by an independent controller. The failure to detect these links hampers the SDN controller from attaining a comprehensive and full overview of the network. Consequently, this poses challenges for the controller in efficiently overseeing and optimizing the network's routing decisions.

Regarding the problem, in [16] instead of proposing a new protocol, the SNMP and OFDP protocols are leveraged to obtain an unified view of the network. An application designed to acquire topology and device state information from legacy network devices is deployed. The application interacts with the SDN controller using the northbound APIs to access the internal topology database of legacy devices. To collect data from the legacy devices, the controller employs SNMP requests and traps messages via the southbound interface using the SNMP. For this approach to be a successfully one, legacy switches must have a pre-established SNMP communication channel with the SDN controller. With the data collected from the OFDP and SNMP messages and traps, the controller has all the information needed to determine all the existing links within the network devices, both legacy and SDN.

The work realized in [17] proposes a completely new mechanism, called Hybrid Domain Discovery Protocol (HDDP), that enhances topology discovery services, enabling the identification of the entire network topology, both legacy and SDN-based devices, in both wired and wireless scenarios, with minimal control message overhead. HDDP utilizes a network exploration model based on flooding to identify non-SDN devices. Furthermore, it introduces a lightweight agent on switches that implement HDDP, facilitating communication of topological information to the SDN controller. Following this, the researchers in [18] improved the HDDP support for diverse wireless networks, now being capable of detecting both unidirectional and bidirectional links between wireless devices.

The proposed network architecture in [19] aims to create a spontaneous wireless hSDN network specifically designed to address challenges in dynamic and heterogeneous environments. One of the key aspects it focuses on is the topology discovery within such networks.

To adapt to diverse wireless technologies, the architecture integrates network interfaces as OF ports. This allows seamless coexistence of devices with different radio access technologies on the same network path, enriching the SDN controller's insights. A monitoring tool continuously assesses communication links and hSDN devices, contributing to real-time adaptation of the network topology.

VI. CONCLUSION

This paper explores Hybrid Software-Defined Networks (hSDN), aiming to seamlessly integrate a diverse range of network devices, including traditional and SDN-enabled components, whether wired or wireless. This coexistence strategy harnesses the combined strengths of these paradigms, providing a versatile and adaptable network infrastructure.

In the context of overcoming challenges, protocols such as OFDPx, eTDP, LADP, and HDDP have been introduced to address specific aspects of the intricate topology discovery landscape. However, they grapple with limitations related to scalability, control plane complexity, and the comprehensive identification of diverse links within the network. While these protocols make valuable contributions to topology discovery, none singularly offers a holistic solution covering all facets of the challenge.

A pivotal aspect of enhancing network efficiency and management in hSDNs involves bringing all these diverse devices under the unified management of the SDN controller. The ongoing dynamic nature of modern networks, featuring various nodes like wired, wireless, mobile, and immobile devices, presents continuous complexities.

Continued research and innovation are imperative to developing more comprehensive solutions. These solutions should not only refine existing protocols but also explore new approaches, ultimately establishing a robust and adaptive topology discovery and management framework for hSDNs.

A. FUTURE WORK

Moving forward, an area of significant focus involves the integration of legacy switches, capable of generating LLDP packets themselves, within the SDN controller's control plane. The current approach uses pre-established SNMP sessions be-

tween the controller and legacy switches, treating the legacy switch as a network node. However, due to the inherent limitations of OFDP in discovering links with legacy devices, the controller struggles to identify to which links the legacy switch posses.

The proposed solution suggests utilizing LLDP packets originating from legacy switches. When an SDN switch receives an LLDP packet from a legacy switch, it forwards it to the SDN controller. The controller processes the packet, identifies its origin, through the pre-established SNMP connection with the legacy switch, and establishes the corresponding link.

GLOSSARY

API	Application Programming Interface
BDDP	Broadcast Domain Discovery Protocol
eTDP	Enhanced Topology Discovery Protocol
HAL	Hardware Abstraction Layer
HDDP	Hybrid Domain Discovery Protocol
hSDN	Hybrid Software Defined Network
ITS	Intelligent Transportation Systems
LADP	Lightweight Automatic Discovery Protocol
LDS	Link Discovery Service
LLDP	Link Layer Discovery Protocol
LLDPDU	Link Layer Discovery Protocol Data Unit
MIB	Management Information Database
NETCONF	Network Configuration
OBU	On-Board Unit
OF	OpenFlow
OFDP	OpenFlow Discovery Protocol
REST	Restful Application Programming Interface
RSU	Road Side Unit
SDN	Software Defined Network
SNMP	Simple Network Management Protocol
TLV	Type-Length-Value
UDP	User Datagram Protocol
VANET	Vehicular Ad-Hoc Network
VTEP	Virtual Tunnel Endpoint

REFERENCES

[1] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[2] F. Bannour, S. Souihi, and A. Mellouk, "Distributed sdn control: Survey, taxonomy, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 333–354, 2018.

[3] S. Badotra, "A review paper on software defined networking," *International Journal of Advanced Computer Research*, vol. 8, 03 2017.

[4] Software-defined networking: The new norm for networks. Online; accessed January 2024. [Online]. Available: https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks/

[5] O. N. Foundation. (2015, 03) Openflow switch specification. Online; accessed January 2024. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[6] Y. Lu and S. Qian, "Research on the theory of snmp and the technology of snmp programming," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5, 2010, pp. 23–25.

[7] GeeksforGeeks. (2022) Types of software defined networks implementation. Online; accessed January 2024. [Online]. Available: https://www.geeksforgeeks.org/types-of-software-defined-networks-implementation/

[8] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: A survey of existing approaches," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[9] Sandhya, Y. Sinha, and K. Haribabu, "A survey: Hybrid sdn," *Journal of Network and Computer Applications*, vol. 100, pp. 35–55, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S108480451730317X

[10] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable openflow-sdn flow control: A survey," *IEEE Access*, vol. 7, pp. 107 346–107 379, 2019.

[11] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in software defined networks," in *2014 8th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2014, pp. 1–8.

[12] H. Tong, X. Li, Z. Shi, and Y. Tian, "A novel and efficient link discovery mechanism in sdn," in *2020 IEEE 3rd International Conference on Electronics and Communication Engineering (ICECE)*, 2020, pp. 97–101.

[13] I. A. Salti and N. Zhang, "An effective, efficient and scalable link discovery (eesld) framework for hybrid multi-controller sdn networks," *IEEE Access*, vol. 11, pp. 140 660–140 686, 2023.

[14] L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "etdp: Enhanced topology discovery protocol for software-defined networks," *IEEE Access*, vol. 7, pp. 23 471–23 487, 2019.

[15] Y. Jia, L. Xu, Y. Yang, and X. Zhang, "Lightweight automatic discovery protocol for openflow-based software defined networking," *IEEE Communications Letters*, vol. 24, no. 2, pp. 312–315, 2020.

[16] F. Kuliesius and M. Giedraitis, "Sdn/legacy hybrid network control system," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019, pp. 504–509.

[17] J. Alvarez-Horcajo, E. Rojas, I. Martinez-Yelmo, M. Savi, and D. Lopez-Pajares, "Hddp: Hybrid domain discovery protocol for heterogeneous devices in sdn," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1655–1659, 2020.

[18] I. Martinez-Yelmo, J. Alvarez-Horcajo, J. A. Carral, and D. Lopez-Pajares, "ehddp: Enhanced hybrid domain discovery protocol for network topologies with both wired/wireless and sdn/non-sdn devices," *Computer Networks*, vol. 191, p. 107983, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621001110

[19] D. Sousa, S. Sargento, and M. Luís, "Hybrid wireless network with sdn and legacy devices in ad-hoc environments," in *2022 13th International Conference on Network of the Future (NoF)*, 2022, pp. 1–9.