IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 January 2023, accepted 22 January 2023, date of publication 6 February 2023, date of current version 1 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242687

TOPICAL REVIEW

Multi-Domain Federation Utilizing Software Defined Networking–A Review

MOHAMMAD HASSAN, MARK A. GREGORY, (Senior Member, IEEE), AND SHUO LI, (Member, IEEE)
School of Engineering, RMIT University, Melbourne, VIC 3000, Australia
Corresponding author: Mohammad Hassan (mohammad.hassan@rmit.edu.au)

ABSTRACT The global growth of the Internet continues with an increase in accessibility, connected users and new applications in emerging fields like eGames, augmented and virtual reality. Technologies that improve reliability whilst reducing cost and latency are being developed. However, before the Internet can sustain the growth in utilisation and new applications there are challenges to overcome. One of the key challenges remaining is to develop new mechanisms for intra and inter-domain federation that facilitate sharing of reachability and routing information. Software Defined Networking (SDN) is being introduced to address control and management challenges at domain boundaries. This paper reviews the current state of domain federation and discusses the potential approaches that utilise SDN for a next-generation solution.

INDEX TERMS Border gateway protocol, future internet, federation, autonomous system, software-defined networking.

I. INTRODUCTION

The evolution of the Internet is redefining the way that we utilise technology to create and consume information. The evolutionary process involves developing new platforms to increase the efficiency of the Internet, support new applications and services and improve information security. The Internet today is global, complex and operates utilising a range of technologies, protocols, devices and systems that adhere to standards set by the international standardisation bodies. The global roll out of new technologies has become increasingly difficult to manage and often means the standards require backwards compatibility at the expense of the efficiency gain promised by the introduction of the new technology.

The Internet is composed of Autonomous Systems (AS) that are a set of Internet Protocol (IP) prefixes for one or more networks that are managed by a single organisation or entity. AS are controlled by the administering organisation, which sets internal routing, rules and policies. The Border Gateway Protocol (BGP) is a standardised protocol that is used to exchange routing and reachability information among the border routers in different AS domains. BGP uses the routing information to maintain a local database of network reachability. BGP then utilises this data to construct a graph of AS connectivity that is used to identify routing paths and to enforce policy at the AS gateway. BGP version 4, the current version, was published as RFC 4271 in 2006 [1] and updated eight times over the past sixteen years. Despite its widespread use, BGP is known to have a number of security concerns [2] and can suffer from slow convergence [3]. Another flaw in BGP is a lack of assurance when validating authorisation for network layer reachability notification messages. BGP's original design did not include the security mechanisms that would prevent the Internet being disrupted either accidentally or by malicious act. To address this problem, a distributed repository system known as Resource Public Key Infrastructure (RPKI) was introduced. However, validation necessitates additional AS implementation stages, which may result in performance and compatibility issues for legacy network infrastructure.

The Internet's current issues are a natural consequence of its evolving architecture. The Internet commenced as a single data communication link and it is now a complex

The associate editor coordinating the review of this manuscript and approving it for publication was Chakchai So-In.

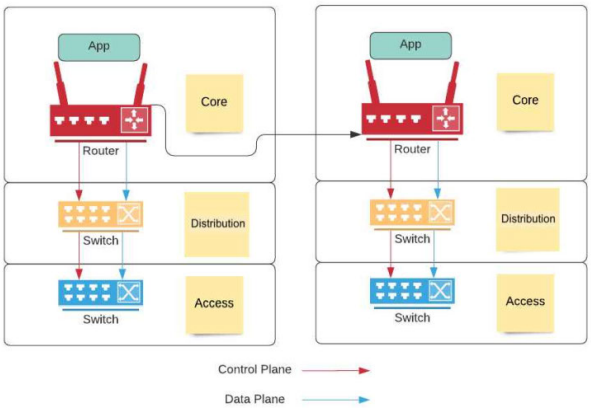

FIGURE 1. Traditional networking.

global network. Due to the increased utilisation and new applications integrated service platforms have been developed and introduced to support IoT, Big Data, AI, and cloud services [4]. However, the existing hardware-oriented, static, and manually-operated network environment does not provide the capabilities and flexibility needed to support increased utilisation and new applications. The Internet's classical architecture, which the SDN paradigm [5] is now being applied to, has become etched in stone, a so-called Internet ossification [6]. SDN is a cutting-edge networking paradigm that has introduced concepts and capabilities that can make networks flexible, reliable and programmatically supports the roll-out of new network management and control applications and services. SDN manages message flows in the data layer using the standardised OpenFlow protocol between SDN controllers and the data flow devices. SDN [7] entices network designers to be rethink and re-image aspects of the Internet that have remained static for decades. It should be noted that it is incorrect to use the terms OpenFlow and SDN interchangeably.

While the first targets of SDN were the cloud and campus networks, we should not ignore its wide-ranging deployment, particularly from the Future Internet (FI) Research & Development (R&D) perspective. SDN's deployment has provided programmable interfaces that enable enhanced automation in provisioning network services. Service providers in general and National Research and Education Networks (NRENs) in particular are also dealing with SDN technologies being incorporated into new deployments [8]. SDN can offer various benefits that can efficiently address cybersecurity issues for cloud-based IoT devices [9]. Within the servers with limited resources, SDN can also be used to manage the diversity of service requests [10].

Employing SDN at an Internet exchange point can improve management and control by using programmable control methods and serving as a Software-Defined eXchange (SDX) centre to handle a large number of advertisements in an adaptive manner. In recent experiments [11], SDN techniques have also been shown to increase the control of BGP routing and improve performance. However, due to BGP's global adoption and the resulting political, technical, and economic obstacles, deploying competing protocols or modified versions of BGP is challenging. As a result, any advancements and proposed solutions should be compatible with the legacy BGP. In light of this, SDN principles have been advocated to be used to overcome BGP weaknesses.

SDN is increasingly being implemented over heterogeneous, multi-technology, and large-scale networks by carriers [12], a key remaining challenge is to harness the flexible and programmatic SDN paradigm between AS domains. Domains are controlled and operated independently, which could restrict the potential for new federated control and routing mechanisms. New domains, as it turns out, do not come into being with instant interoperability and federation. The new domain is federated into the Internet through the manual setup of its border router and the implementation of the domain control and routing policies.

An SDN based approach for multi-domains would require a control channel between domain gateways for inter-domain SDN federated communication. The advantages of implementing SDN into multi-domain networking compared to traditional methods are its ability to provide policy-driven network monitoring, agility, traffic management and control, and apply network automation. Multi-domain SDN could make it possible to connect different SDN domains automatically, bring about domain interoperability, and improve the offering of cross-domain services.

By enabling a software-centric, dynamic, automated, and intelligent network environment, multi-domain SDN [13] can be deployed to create a next-generation federation framework. The framework should reduce operating costs and enable the flexible management of traffic flows. Inter-domain traffic will be enhanced by removing the control and management complexity of legacy systems and by introducing advanced network service management, including flexible peering arrangements, bandwidth on demand, and real-time provisioning [14].

The reachability of network nodes in SDN based domains is a current research and development focus. The end-to-end access rule provisioning requirements and architectural components have not yet been fully defined. In this paper, we review the SDN literature and present the SDN based multi-domain federation concept, the basis for a multi-domain, vendor-neutral SDN based domain federation and the architectural theory. We also provide an overview of recent developments in federated domains and address research issues and strategies for future implementations.

The remainder of this paper will be structured as follows. The SDN paradigm, architecture and control environment are presented in Section II, followed by different multi-domain communication mechanisms in Section III. Section IV discusses federation, including its advantages, applications, and requirements. Then, in Section V, the building blocks and architecture for SDN based federation are covered. Current SDN based federation test beds are presented in Section VI.

Section VII discusses the opportunities and challenges of SDN based multi-domain federation. Section VIII brings the paper to a conclusion.

II. SDN CONCEPTS

Current networks can be vertically integrated, which increases complexity with the control and data planes bundled together as shown in Fig. 1. Despite their widespread acceptance, conventional IP networks are complex and difficult to manage. Configuring the network utilising vendor-specific commands according to predefined policies is complicated, and needs constant updating to respond to faults, load and route changes. The distributed network control and transport protocols are the key technologies that allow information to travel around the world in the form of digital packets [15].

Understanding the complex nature of existing IP networks is a focus for the research community and industry. With the knowledge gained, new concepts have been proposed for the design of future networks [16]. Some of the innovative concepts have been implemented, including programmable networks [17], Named Data Networking (NDN) [18], and ''HTTP as a narrow waist'' [19].

A. SDN EVOLUTION

From a network architecture perspective, the SDN capabilities and features mentioned are not entirely new. There were previous attempts to encourage the programmable networking approach. The history is divided into three stages [20], each with its own contribution: (1) active networks (from the mid-1990s to the early 2000s), which introduced programmable functions in the network, resulting in greater innovation; (2) control and data plane separation (from around 2001 to 2007), which developed open interfaces between the control and data planes; and (3) the OpenFlow Application Programming Interface (API) and network operating systems (from 2007 to around 2010), which represented the first widest range of open interfaces between the control and data planes. Throughout the history of SDN, network virtualisation has played a key role.

Active networking concept attempts to control a network using software in real-time. SwitchWare [21], [22] is an active network-functioning solution that permits packets to flow through a network and dynamically change network operations. Similarly, software routing suites on traditional PC hardware, such as Click [23], XORP [24], Quagga [25], and BIRD [26], are the basis for extensible software routers utilising programmable network devices. By loading new or changing existing routing software, the behaviour of certain network devices can be changed dynamically.

The last decade has seen an increase in research into how to decouple the data and control planes. Caesar et al. first proposed a Routing Control Platform (RCP) in 2004 [27] in which inter-domain routing is replaced by centralised routing control to reduce the difficulty of fully distributed route computing. The IETF released the Forwarding and Control Element Separation (ForCES) framework in the same year, which separates control and packet forwarding elements within a ForCES network [28], [29], [30], [31]. In 2005 Greenberg et al. suggested a 4D approach [32], [33], [34] implementing a blank slate concept of four planes for the entire network architecture. The planes are, respectively, ''decision,'' ''dissemination,'' ''discovery,'' and ''information'' arranged from top to bottom. In 2006, the Path Computation Element (PCE) architecture was introduced to measure label switched paths separately from real packet forwarding within Multiprotocol Label Switching (MPLS) and Generalised Multiprotocol Label Switching (GMPLS) networks [35]. In 2007, Casado et al. introduced Ethane, where simple flow-based Ethernet switches are complemented by a centralised controller to handle flow entry and routing [36], [37], [38], [39]. Commercial networking devices also adopted the idea of separating the control and data planes. For example, the control plane is decoupled from the data plane and modularised in the Cisco Aggregation Services Routers (ASR) 1000 series routers and Nexus 7000 series switches, allowing for the coexistence of an active control plane instance and a standby instance for high fault tolerance and transparent software upgrades.

Compared with traditional networks, the value of SDN lies in that it offers programmability as well as the decoupling of the control and data planes. In particular, SDN provides a flexible approach to introducing APIs to the controllers. In addition, SDN clearly distinguishes the control and data planes within the network architecture. The plane separation means that network service control and monitoring functions can be added, removed, updated and operated independently of the data plane flow devices, thereby without impact on the traffic flows. SDN has significantly improved the opportunity for dynamic and flexible traffic control and management [40].

B. SDN BUILDING BLOCKS

The Open Networking Foundation (ONF) describes SDN [41] as the ''physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.'' SDN has the following capabilities and features:

1. SDN decouples the control plane from the forwarding plane [42], utilises OpenFlow for traffic flow control messages passed between the SDN controllers and infrastructure layer data flow devices, and provides a northbound API to a programmatic application layer. The SDN paradigm facilitates network services, hosted in the SDN controllers, that manage and monitor the infrastructure layer devices utilising standardised and non-standardised protocols.

2. In SDN the packet forwarding decisions are based on flows rather than destinations. A flow is usually defined by a collection of packet field values, which serve as a criterion for the match (filter) and a collection of actions (instructions). SDN categorises a flow as a packet series between a source and a destination. At the forwarding devices [43], [44] all

TABLE 1. Legacy networking versus SDN.

	Traditional Networking	SDN
Attributes	New protocol per problem, complicated network control	programmability
Configuration	Error susceptible to manual setup	Automated configuration with centralised validation
Performance	Dynamic, global control with information across layers	Limited details and a fairly static configuration
Innovation	Simple implementation of software for new ideas, adequate insulated test environment, fast development with software upgrade	Implementation of hardware difficult for new concepts, restricted testing environment, long standardisation procedure

packets of a flow receive similar service policy. The flow abstraction allows the behaviour of various types of network devices to be unified, including routers, switches, firewalls, and gateways [45]. Flow programming permits unparalleled flexibility, restricted only to the capacities of the flow tables implemented [13].

3. Control functionality is removed from network devices that transition to become simple message forwarding devices. The control logic is transferred to the so-called SDN or Network Operating System (NOS) controller. The NOS is a software platform running on commodity server technology and providing the essential resources and abstractions to facilitate forwarding device programming based on a logically centralised, abstract view of the network. Its function is similar to a conventional operating system.

4. SDN introduces a network architecture that makes it possible to programmatically control rather than configure computer networks. It is a core function of SDN, considered as its primary value proposition.

C. SDN BENEFITS

SDN tackles the challenges of the legacy network architecture as outlined in Table 1. The benefits are derived from enhanced configuration options, increased performance and innovation in network design and operations.

1) ENHANCED NETWORK MANAGEMENT

Network configuration involves a degree of manual processing due to the heterogeneity of network device manufacturers and configuration interfaces. This manual setup process is both repetitive and vulnerable to error. SDN can help remedy this network management problem. In SDN, the integration of the control plane over all types of network devices enables network devices to be configured from a single point, automatically through software control [46]. As such a network can be programmatically configured and optimised dynamically depending on the network status [47].

2) IMPROVED NETWORK PERFORMANCE

One of the main goals of network operations is to optimise the utilisation of the invested network infrastructure. However, maximising network efficiency as a whole was deemed difficult due to the coexistence of various technologies and stakeholders in a single network. Current approaches concentrate on optimising a subset of network outputs, or the quality of user experience for selected network services. Obviously, the strategies employed may lead to suboptimal efficiency, if not conflicting network operations, based on local information without consideration of cross-layers. SDN creates an opportunity for a global change in network efficiency and explicitly offers a global view of the network including data flow control, management and monitoring across the various layers. It follows that new solutions for classical problems, such as data traffic scheduling [48], end-to-end congestion control [49], load-balanced packet routing [50], energy-efficient operation [51], and support for Quality of Service (QoS) [52], [53], can be developed and deployed to verify their effectiveness in improving network performance.

3) THE EMERGENCE OF VIRTUALISATION

SDN is an exciting paradigm used to facilitate hyper-scale Data Centre (DC) management. DCs raise important scalability concerns, especially with the growth and migration of Virtual Machines (VM). Moving a VM and modifying the address table for Media Access Control (MAC) using the conventional network architecture will disrupt user experience and applications. Network virtualisation is a key technology that is used to optimise infrastructure utilisation in hyper-scale DCs. It provides tunnels that can abstract the MAC address from the infrastructure layer, allowing traffic from Layer 2 to run over Layer 3 overlays and simplifying network deployment and migration of VMs [54]. In addition, SDN enables multi-tenant hosting providers to link their physical and virtual servers, local and remote facilities, and public and private clouds to a single logical network. As a consequence, the network operator would have an independent view of each customer. SDN facilitates the orchestration of a virtualisation layer to cloud provider network architectures.

D. SDN ARCHITECTURE

SDN includes three abstractions: specification, delivery, and forwarding. The abstractions are essential tools for architecture design, implementation and future research [55].

The *specification abstraction* will allow a network application to communicate the desired network behaviour without being responsible for enforcing the behaviour itself. This can be achieved through network programming languages as well as virtualisation solutions. These approaches map the abstract configurations represented by the applications, based on a condensed, abstract network model, into a physical configuration presented by the SDN controller for the global network view.

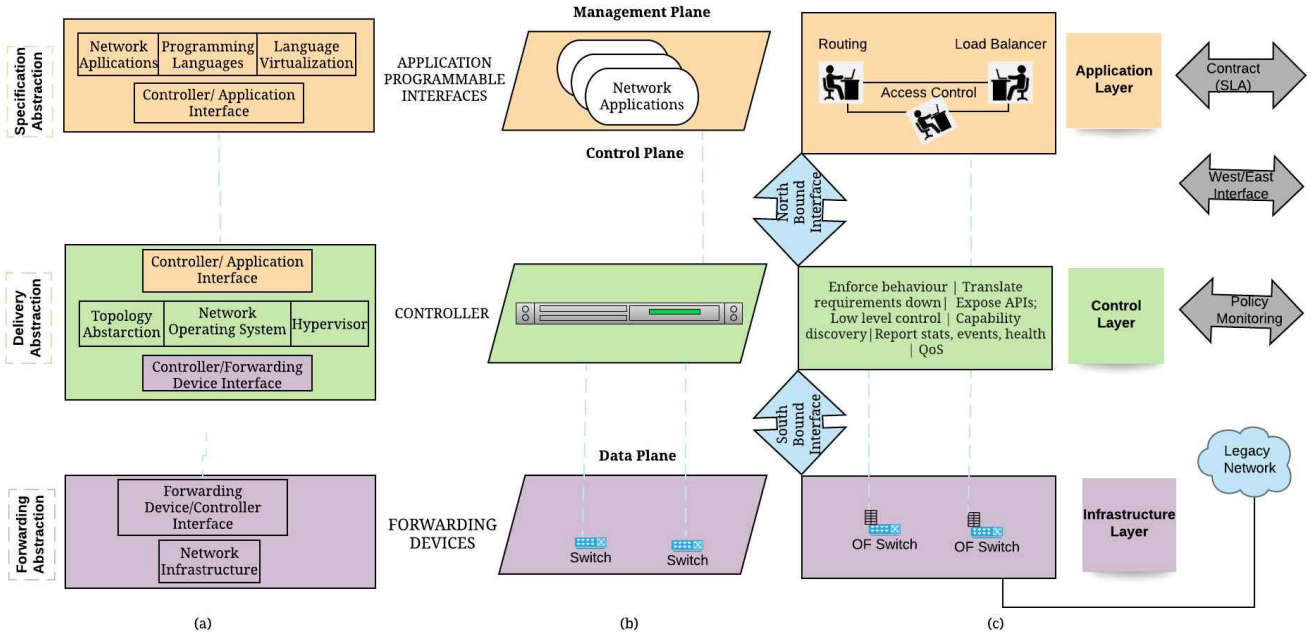

FIGURE 2. SDN in (a) abstraction view, (b) planes and (c) system design architecture.

The *delivery abstraction* shields SDN applications from distributed state complexities, making the distributed control problem into a logically centralised one. Its realisation requires a common layer of distribution, which resides in the SDN NOS. There are two basic functions to this layer. First, it is responsible for installing the control commands into the forwarding devices. Second, it gathers forwarding layer status information (network devices and links) to provide network applications with a global view of the network.

Ideally, the *forwarding abstraction* permits forwarding behaviour that is required by the network applications (the control system) while hiding the underlying infrastructure. OpenFlow is one realisation of an abstraction, which can be used as the equivalent of a ''device-driver.''

The SDN architecture is shown in Fig. 2. Key features of this architecture include:

Management Plane or Application Layer. The management plane is the collection of applications that take advantage of the functions that the northbound interface provides to implement network control and operating logic. This includes routing, firewalls, access, authentication, load balancing, and tracking. In essence, a management application defines the policies which are ultimately translated into southbound specific instructions that control the actions of the forwarding devices.

Northbound Interface. Application developers have a northbound interface API accessible from the NOS, that is, a common interface for overlay applications. Usually, an API abstracts the low-level instruction sets used to control or monitor forwarding devices via the southbound interface.

Control Plane or Controller Layer. There are various applications running in the controller and abstracted for the Operation & Management (O&M) as well as maintenance of the entire network. In summary, the SDN controller provides and retains a global view of the whole network from which users can create further applications to enhance the efficiency of the network and resource utilisation.

Southbound Interface. The forwarding devices instruction set is specified using the southbound interface API. In addition, the southbound interface also defines the communication protocol between the forwarding devices and control plane elements. This protocol formalises how elements of the control and data plane interact with each other.

Data Plane or Infrastructure Layer. Wired cables or wireless radio channels interconnect the forwarding devices. The network infrastructure includes the interconnected forwarding devices, representing the data plane.

Forwarding Devices. The data plane forwarding devices are network devices that perform a series of elementary operations. The forwarding devices have well-defined instruction sets, e.g., flow rules, used to take actions on the incoming packets that may include port forwarding, packet dropping, interaction with the controller, header rewriting or packet encapsulation.

The SDN controllers interact with the underlying data layer forwarding devices to optimise traffic flows and to manage and monitor the network. The control layer facilitates business logic, applications and services that reside in an application layer connected to the controller via the northbound interface API. Controllers can utilise east-west interfaces as

a bridge to adjacent controllers or other network devices. The architecture implemented to support network control can affect localised controller performance and scalability [56].

E. CONTROL ENVIRONMENT

SDN has been applied to intra and inter-domain networking scenarios. The SDN network architecture implemented to control, manage and monitor domains can be centralised or distributed and both can utilise a hierarchy of controllers. SDN has both physical and virtualised control planes. In this section the SDN controller distribution is discussed.

1) SINGLE CONTROL PLANE

Early SDN implementations were developed for single controller single domain settings principally situated in a data centre. This type of control plane is referred to as the single control plane and utilised a basic NOX controller [44]. To control, manage and monitor the domain network, Ethane, an early implementation, adopted a single controller approach. Ethane [38] implementations were designed to control, manage and monitor an estimated 10,000 or more data flow devices using a single controller. For small networks, this solution was suitable but it is unusable for large deployments. To reduce controller loads, the single controller technique utilises two approaches:

- **Hardware or Multi-core Controller.** When SDN was implemented, one fundamental problem was how to identify routing solutions using a single controller. If the controller lacks the capability to manage the flow requests, it will become a bottleneck for the entire network. For this reason, a range of proposals were put forward to improve the single controller's processing capability, such as Beacon [57] and McNettle [58]. Beacon offers a platform to control network devices and achieves high performance by using multi-core processing. In addition, the output can be scaled-up linearly with the number of processing cores. For example, Beacon with 12 cores was able to handle 12.8 million flow requests per second.
  However, the simplicity of a centralised controller comes at the expense of the control plane having minimal scalability. McNettle is designed to solve this issue as an extensible control device. The control capability can be expanded by adding handlers and background programs that utilise a multi-core processor. Using a single 46-core processor, McNettle was capable of serving up to 5000 switches while achieving a throughput of over 14 million flows per second. Maestro [59] provides the application programmers with a basic single-threaded programming model. In addition, parallelism optimisation techniques are used to increase a controller's throughput. Using an eight-core controller, Maestro can handle 600,000 flow requests per second. Although this approach can boost the capability of a single controller, one controller is inadequate in some network scenarios. For SDN, the new flow routing paths must be determined by the controller. The time it takes for the controller to assign a routing path when it receives a flow request should be minimal.
- **Overhead Reduction.** The second approach aims to minimise controller load, as demonstrated by proposals such as DIFANE [60] and DevoFlow [61], by expanding the function of the switch data plane. DIFANE partially uses intermediate switches called authority switches to make transmission decisions instead of relying solely on the centralised controller. In DevoFlow a similar method is also suggested with the collection of different flows to be guided to the controller, while switches manage the other flows. In data centres a centralised controller has limited ability to adjust to changes in the inter-network load.

In the case of the Internet, a single controller is not appropriate due to the broad geographical range, network complexity and the need for organisations to control and manage AS domains. Remote flow request transmission increases delay. SDN based networking implementations today generally utilise multiple controllers. The only implementations that would not use a controller hierarchy are small networks.

2) THE MULTIPLE CONTROL PLANE

Multi-controller implementations are an unavoidable and rational solution that provides reliability, scalability, and security. A single controller within the network is a single point of failure and a promising option for attackers. A malicious or malfunctioning controller can also easily disrupt the entire network. Hence the multi-controller configuration reduces risk.

Multiple controller implementations have key issues. The first challenge is the requirement that the controllers have a single network-wide vision. This problem is not solved using static configurations, and it is possible to have individual controllers with a much higher load than other controllers in the network. The routing announcements, path identification and linkage between the control plane and the forwarding plane should be automated. The second challenge is to find an optimum number of controllers to ensure there is a linear scaling up of the SDN network. The final challenge is how to synchronise network operation and routing. Most methods use localised solutions to control, manage and monitor the local neighbourhood.

There are two distinct methods of implementing the multi-controller approach to control, manage and monitor the network as outlined in Table 2.

- **Logically centralised.** The logically centralised controller hierarchy exchanges information to create a coherent view of the entire network. To centralise the logic, Gao et al. [62] used three approaches; a distributed file system, a distributed hash table, or a precalculation of all possible combinations. The controllers maintain a highly consistent network-wide view. The

TABLE 2. Centralised vs. distributed SDN controller.

	Centralised Control	Distributed Control
Pros	Single point of management	Scalable to large networks
	Better control over the consistency of the network state	More responsive to handle local network event
	Global view of the network is great idea for network-wide states and decisions	Easily adaptable to the requirements of users and applications
	More Granular security	Resilient to failures
	Lower capital and operational expenditure costs	Support controller clustering
Cons	Not scalable	Expensive to manage-protocol experts
	Higher overload	Requires a consistent network-wide view in all controllers
	Significant challenges in terms of flexibility and robustness for large networks	It can lead to unequal distribution of load between controllers
	Extremely vulnerable to threats and disruptions	It is difficult to find the optimum number of distributed controllers
	Lower capital and operational expenditure costs	Support controller clustering
	The dynamics needed to support versatile communication services are missing	There remains the need to synchronise the overall and dispersed events in order to give the local a global view

network-wide view is synchronised, however, this is a complex process. If a controller's local view changes, the controller synchronises the changed state information with the other controllers in the hierarchy. The information exchange or state synchronisation between controllers absorbs network resources and a challenge is to minimise the resulting network load.

Onix [63] focuses on providing generic distributed state management APIs for control logic overlay applications. Onix is used with a global network view.

Hyperflow [64] suggest logically centralised control with a scalable controller hierarchy. A NOX based implementation was described where the controllers can be deployed on demand by network operators. HyperFlow permits the controllers to maintain a network-wide view by passively synchronising controller state events.

DISCO [65] is an open and extensible SDN control plane architecture designed to tackle the distributed and heterogeneous characteristics of wide-area networks and modern overlay networks. DISCO is implemented using Floodlight, an open-source OpenFlow enabled controller and consists of two parts: an intra-part where each controller manages its network domain, and an inter-part which manages communication with other DISCO controllers to ensure end-to-end network services through a lightweight control channel.

OpenDaylight (ODL) (Jahan et al., 2014; Medved et al., 2014) [66] allows a conceptual or physical division of the network into separate slices or tenants. ODL implementations may include a combination of the conceptual and physical sub-divisions. The ODL design includes a management entity that allocates resources to the network slices or tenants.

The OpenContrail Controller is logically centralised and provides network management, monitoring and analytics. OpenContrail is described as a cloud-network virtualisation tool. There are two key components of the OpenContrail system: the OpenContrail Controller and the OpenContrail vRouter.

- **Distributed.** The distributed controller hierarchy introduces a physically distributed control plane state and logic; thus the global view becomes an overlay management and monitoring application. Due to the frequent network changes that occur today in the complex global network, network state synchronisation may lead to network overload and control state inconsistency may occur. The controllers manage the routing and path announcements for a local domain. There are two implementation approaches for distributed controller hierarchies: horizontal and vertical.

  Kandoo [67] is a symbolic control plane with a hierarchical architecture. Kandoo uses two control layers, i.e., root (top layer) controller and local controller (bottom layer). Kandoo is used by network operators to deploy on-demand local controllers that alleviate the load on root controllers.

  Levin et al. [68] found that utilising a centralised SDN control plane degrades the efficiency of many applications and the study identified that the control plane state and logic should be physically distributed. Levin et al. also showed that a controller can derive a non-optimal but reasonable flow path, even if the network view is limited in scope.

  Tam et al. [69] and Schmid and Suomela [70] carried out further study in this field. Tam et al. proposed multiple independent controllers with each controller managing a segment of the network. Schmid and Suomela moved this approach forward with the introduction of improved routing algorithms.

  ElastiCon [71] proposed a controller pool that grows or shrinks dynamically according to traffic conditions and workload.

  Pratyaastha [72] proposed an Efficient Elastic Distributed SDN Control Plane; a novel method for assigning SDN switches and SDN device state partitions to distributed controller instances.

F. SDN SECURITY ISSUES

Applying SDN to multi-domain networks introduces security challenges that have been highlighted in the literature [73], [74], [75], [76]. The overview of SDN security issues is shown in Fig. 3.

Some of the security challenges arise in the control and application planes. Applications that can be subverted and used to insert malicious configurations into network devices without authentication can decrease network capacity and

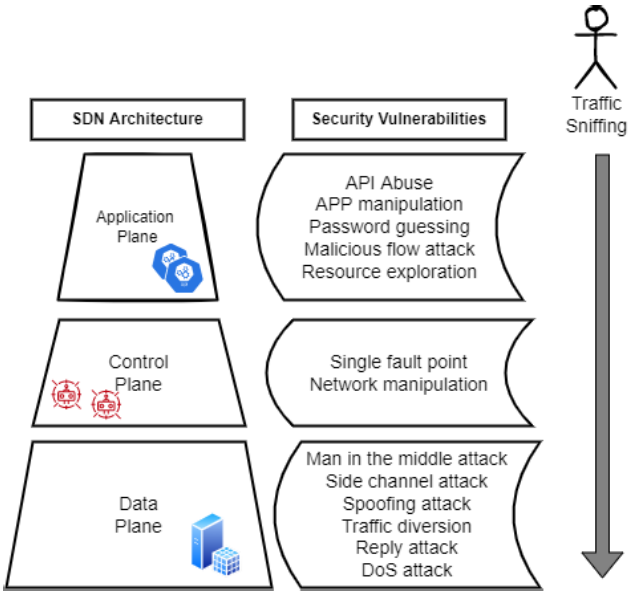

FIGURE 3. SDN Security issues.

stability, possibly resulting in network failure. Loss of traceability and transparency of application flows can cause network debugging difficulties. In multi-domain SDN, application flows and network state monitoring and auditing can help track and replay network state or debug a network fault. Information on network activity can also be used to identify patterns of attack [77]. The control plane also exposes network-wide resources to overlay applications, opening a door for malicious applications. Additionally, adopting a single controller may result in a single point of failure that could become an attractive target for Denial of Service (DoS) attacks.

When the configuration-complex, Transport Layer Security (TLS) protocol is not implemented, the lack of an authenticated controller-switch communication channel may result in more serious security threats. Adversaries may start a man-in-the-middle attack by seizing all messages between controllers and switches [78], [79]. Additionally, malicious switches may also initiate spoofing attacks by faking identities, e.g., IP addresses, that could lead to DoS or DDoS attacks [80], [81].

In recent years several proposals have been made [82], [83], [84], [85], [86], [87], [88], [89], [90] to resolve the multi-domain SDN security issues. The proposals tackle particular security issues. In particular, current work providing application flow authentication or flow-safe restriction extend secure modules on a controller rather than support a monolithic secure module in a multi-controller environment. Existing role-based access control systems need to be fine-grained on network-wide resources. The optional TLS protocol and other cryptography-based authentication protocols allow multiple interactions (also called multiple passes) to create a communication channel for controllers. In a network with a physically decentralised control plane, merely combining existing schemes does not solve the common security problems effectively as the secure modules have to operate seamlessly between several controllers. A universal security approach that resolves the common security challenges has not been proposed in the literature.

III. MULTI-DOMAIN COMMUNICATION

In an inter-network spanning multiple administrative domains, a logically decentralised control plane is needed. While the domain operators are not likely to agree to unified control, it may be reasonable to provide a certain level of sharing (e.g., to ensure that service level agreements are met for traffic flowing between and transiting domains). Some architectures on distributed control planes with a logically centralised approach such as Onix, Hyperflow, and Elasticon are currently unable to handle inter-domain flows between SDN domains. According to Egilmez (2014) [91] the fully distributed SDN controller architectures, both vertical and horizontal approaches, may be used for multi-domain SDN communication. A multi-domain SDN architecture refers to the architecture of a network linking multiple SDN domains. SDN domain refers to the administrative SDN domain, which can be a sub-network in a data centre network, or a carrier or business network, or an AS. A multi-domain architecture allows various functional domains to be generated within the SDN model that can be managed by different parties, such as the Internet. Such administrative domains will regulate entirely which information and resources they are presenting to others. Three methods for modelling Controller to Controller (C2C) relationships were suggested for multi-domain deployment [92].

The tree-based model (the vertical approach) has tree-like controllers where parent controllers can delegate some roles and responsibilities to their subordinates but maintain complete visibility of all actions and states. Multi-domain services are coordinated in this configuration by requests being sent to the parent controllers by their subordinate controllers when changes to operations occur. Control tasks are allocated to different controllers in this deployment model, based on selected parameters such as network view and locality requirements. Thus, the controller manages local events and is lower in the hierarchy, and the higher level manages global events.

The chain-based configuration model (horizontal approach) has logically configured controllers to communicate directly only with their neighbours, and communication with other controllers in other domains is enabled by the neighbouring networks relaying requests to the next-hop controller. Multiple controllers are arranged in a flat control plane, in which each controls a sub-set of the network switches. This can be achieved either with replication of the state or without replication of the state. There is no reciprocal regulation of administration within these various administrative realms. The domain is managed by its collection of controllers, and requests for multi-domain services depend on the functionality. To do this, an east/west-bound interface

communication protocol must be introduced, present and accessible to controllers within each domain.

Hybrid setup is a mixture where the controllers may also provide a certain level of logical decentralisation. An interesting type of proxy controller, called FlowVisor [93], can be used to introduce a level of network virtualisation to SDN networks and allow multiple controllers to manage overlapping physical switches simultaneously. It was initially designed to allow experimental research to be carried out alongside production traffic on deployed networks, and also facilitates and demonstrates the ease of deploying new services in SDN environments.

A crucial question is how many network abstraction modules can be centralised and configured effectively to serve logically structured control tasks while providing physically distributed protocols at the same time. Consequently, hybrid SDN architectures are being considered as an approach to overcome current challenges. Typically the controllers are arranged utilising a combination of tree and chain structures. Orchestrators in this approach will include standard interfaces, frameworks and policies to manage and communicate in distributed environments with the control planes, and will provide high-availability and fault tolerance capabilities.

The separation of SDN controllers into disjoint peer domains may bring benefits:

- Controllers can come from various suppliers who have not achieved full interoperability
- Controllers or underlying infrastructure may be owned or run by specific administrative bodies
- Controllers can have specific hardware or device features
- Network node count or geographic position scalability, including the difference between WAN and LAN

In general, a service can cross any number of forwarding planes, and thus many Network Control Domains (NCDs), even domains that are not SDN enabled, so-called nSDN. Such services include cooperation between neighbouring controllers, as well as management, power, or signalling with nSDN. As controllers interact through various administrative boundaries, which may also be company or operator boundaries, there is a need for cooperation to determine how network control, management and monitoring is to occur. This may be simply a decision to provide outward facing gateways that have standardised interfaces and AS numbers. Controllers may make available the following information to authorised neighbours:

- Adjacency and capability discovery of SDN controllers
- Forwarding plane neighbours and topology discovery to the extent agreed in policy State and attribute information, including the ability to subscribe to state and attribute change notifications, as agreed in the policy
- Forwarding-relevant information such as accessibility at one or more levels
- Path computing information such as route cost, security or restoration policies
- Other information such as Operations, Administration, and Maintenance (OAM) setup, QoS assessment and reporting, billing usage details.

A. RELATED WORK

SDN researchers are exploiting the possibilities of using OpenFlow-enabled SDN to build programmable WAN architectures or allow expressive policies in inter-domain routing. An SDN-IP gateway [94] could be configured to peer with other SDN-IP gateways across the WAN in order for a SDN based domain to connect to the Internet. This technique utilises the BGP protocol, which is a legacy approach and does not fully exploit the SDN paradigm. Research on this problem [95] recommend modified BGP protocols to improve the reachability and routing update process, while other study suggested reachability and routing information exchange could occur by utilising links between private and secure east-west SDN controller interfaces [96]. Several works, such as SDNi [97], Google [98], Microsoft [99], Noise Lab [95], and ON.Lab (Open Networking Laboratory) [42] will be addressed in the following section to provide examples of research into new network architectures.

1) SDNi

The inter-SDN domain protocol (SDNi) can be used as an interface between the controllers in different SDN based domains. The controllers are able to share topology, routes and path information, energy consumption and QoS-related policies and information. In addition, the SDNi protocol permits sharing information on reachability and facilitates end-to-end Service Level Agreements (SLA) over heterogeneous networks. To ensure the extensibility of its transport mechanisms and syntax, SDNi still lacks a semantic network model. Ideally, a completely automated service, from negotiation to order, should be assisted by delivery confirmation and charging [100].

2) GOOGLE B4 (GLOBALLY DEPLOYED SD-WAN)

While utilising conventional WAN architectures, Google B4 was motivated by the finding that Google was unable to achieve the degree of the scale, fault tolerance, cost efficiency and control needed for its network. Google's networking requirements included meeting dynamic demands for bandwidth, reducing website page display times, end application controls, and cost sensitivity. An SDN based approach was used to meet the requirements identified. Google adopted commodity servers and devices rather than utilise traditional vendor products. The SDN ecosystem permits the rapid iteration of novel protocols, ii) simpler testing environments, iii) improved capability planning, and iv) simpler management through a fabric-centred, rather than router-centred, WAN view. Google created an internal distributed data centre network called ''G-Scale'' with the SD-WAN solution. The specification for the B4 sites involves the hardware layer forwarding traffic without the intervention of a complex

control program. The site controller layer is composed of Network Control Servers (NCS) which hosts both OpenFlow controllers and applications for network control.

3) MICROSOFT SWAN (SOFTWARE-DRIVEN WAN)

Microsoft chose to develop a new WAN architecture with the goal to improve throughput and to promote flexible network-wide sharing. SWAN aims to provide:

- Services, excluding interactive, that inform the SD-WAN controller of current traffic demands between DC pairs.
- The controller, which has an up-to-date global view of the topology and network traffic demands, monitors traffic volumes and route utilisation.
- The controller changes the forwarding statuses of flow devices explicitly based on current network demands.

Microsoft's design involves the collective deployment of service hosts and brokers to estimate current network traffic demands and this information is used to manage the network utilising distributed controllers.

4) SOFTWARE-DEFINED EXCHANGE POINT

Chung et al. [101] proposed to change the delivery of wide-area traffic by developing, prototyping and implementing a Software-Defined eXchange (SDX) that addresses four challenges: i) convincing applications, ii) programming abstractions, iii) scalable operation and iv) practical implementation. Virtual SDX switch abstraction allows each AS to run SDN based applications defining versatile traffic drop, change, and forwarding policies, and the SDX can be used to merge several AS policies into a single, coherent physical forwarding policy. The SDX route server enables each user to forward traffic for a prefix to all feasible routes, as it has specific features such as overriding default BGP routes and integrating SDX with existing infrastructure. The SDX controller implementation has two main pipelines: a policy compiler and a route server [95].

5) ONOS

To address the need for an OpenFlow-based carrier-grade SDN controller, Stanford University's ON.Lab leads the development of Open Network Operating System (ONOS) [102]. ONOS [103], followed in the footsteps of previous SDN controllers like Onix but was released as an open-source project that relies on the SDN community to contribute, review, and analyse its use and operation. ONOS offers a global view of applications on the network, which is logically centralised even though network state information is spread among the controllers. ONOS abstracts device characteristics such that the core operating system does not need to be aware of the particular protocol used to control a device. It offers an open framework by managing network resources and offering high-level abstractions and APIs that simplify the development of creative and beneficial network applications and services that operate across a wide range of hardware. ONOS adopts a distributed architecture for high capacity availability and scale-out. Global Network View, fault tolerance, low-latency, optimised data model, in-memory topology view, and event notifications are some of the key features of ONOS.

B. LIMITATIONS OF CURRENT WORK

Work has been carried out to consider how BGP might be updated to accommodate SDN based networking. Le et al. suggested an SDN Federation Protocol (SFP) [73] utilising a publisher-subscriber model approach to avoid excessive combinations of packet headers with routing entries. However, the authors did not mention the problems when accommodating non-adjacent network configurations, multiple similar flow queries causing flooding, and accuracy and stability. The current inter-domain SDN based network communication techniques do not address a number of issues, including multi-field prefix matching, end-to-end routing between SDN based domains, and multi-path data transmission [104]. The current research are not applicable to a variety of application domains, including IoT, Big Data, AI, and cloud services, where it is necessary to connect the distributed SDN based domains in order to provide integrated service [105]. The communication between the various controllers of the various SDN based domains was managed by a research project [106] based on a global centralised controller, which allowed the local controllers to quickly exchange control messages with relatively low costs and high data transmission performance by establishing layer-2 (L2) network connections to other controllers. Due to the numerous linkages for the data transfer between global and local controllers, the problems with this approach is that there are significant message overheads and scalability issues. In order to provide low message overheads and high federation scalability, an inter-SDN based network federation scheme [107] was conducted based on the distributed and L2 oriented network architecture using direct and simplified message exchanges between the local SDN controllers (with the global controller excluded). Through the data modelling abstraction of the information from the local SDN based network and the streamlined interactions between controllers using automated setups and specialised application interfaces, this research effort leads to a more effective inter-SDN based network architecture. The main weakness of this study is that it only tested inter-SDN based network federation on a small number of SDN based domains. However, federated SDN based networking should be further explored to examine extensibility across numerous domains.

IV. SDN BASED MULTI-DOMAIN FEDERATION

In the specific context of the SDN initiative, the U.S. Global Environment for Network Innovations (GENI) [108] defines: ''A federation is a collection of agreements between individuals or organisations that represent the policies and conditions under which they trust, collaborate, share resources or participate in other joint activities.'' The federation determines the policies and terms of resource use based on the individuals

and organisations involved interests and needs. The controller hierarchy for intra-domain federations is represented by the multiple control plane communication outlined in Section II.

Multi-domain federation requires participating systems to use a signalling protocol that manages ''inter-work.'' A federation automatically calls for a ''promise'' between the participating bodies to a certain degree. One can think of each network or entity that manages a part of the network as an autonomous agent that makes ''promises'' with the other agents about the type of service they can provide [109]. Agents are autonomous, have a view of their local network and not a worldwide view. The agents can be cooperative or non-cooperative in a distributed environment. Another interesting feature of multi-domain federation is that it usually doesn't include a completely consistent distribution of state knowledge. Indeed, agents often do not know (nor need not know) the exact state or utility function of the other agents. The desired result can be accomplished by the agents autonomous activity as opposed to a central core of information that can be used to micro-manage.

The concept of ''super SDN controllers'' that micro-manage smaller organisations or networks can be equated with the centralised federation mode; unified knowledge base, homogeneous structures, single administrative jurisdictions, and a ''least common denominator'' offer to some degree. The central SDN controller is constrained by the weakest link capabilities and becomes a bottleneck. It is apparent that the design of a distributed structure in which individual controllers can ''promise'' and provide services to other controllers will lead to a much more flexible system, provided it is properly designed. The use of current federation structures is a smart option for the near future. There is a possibility that a limited distributed control approach is not optimal. This may or may not be the case, however, depending on the distributed system architecture. Although a centralised controller that manages everything is conceptually easier to understand, it is not necessarily scalable or operationally functional.

The ''federating'' networks can utilise overlay applications to gain oversight and to control, manage and monitor the federated networks. Overlay and underlay in a virtualised network work together to deliver the outcome. When integrating the two together, a tight coupling or a loosely coupled arrangement can be selected where the ''underlay'' can make clear ''promises'' to the overlay regarding the types of services it can deliver, and it can also inform the ''overlay'' when it should stop providing services. Moving forward new structures for overlay and underlay interactions are anticipated. This does not imply a centralised controller approach nor the development of overlay applications that limit network service operation. It is envisaged that a network where multiple overlays exist that can use the same or multiple underlays providing various types of services and different service APIs, and where they can work together to achieve a shared end-to-end infrastructure. The way to achieve these goals is neither with super controllers nor super APIs. What is needed is a theory-based model that allows independent agents that reside in overlays and underlays to interact with each other to deliver a service.

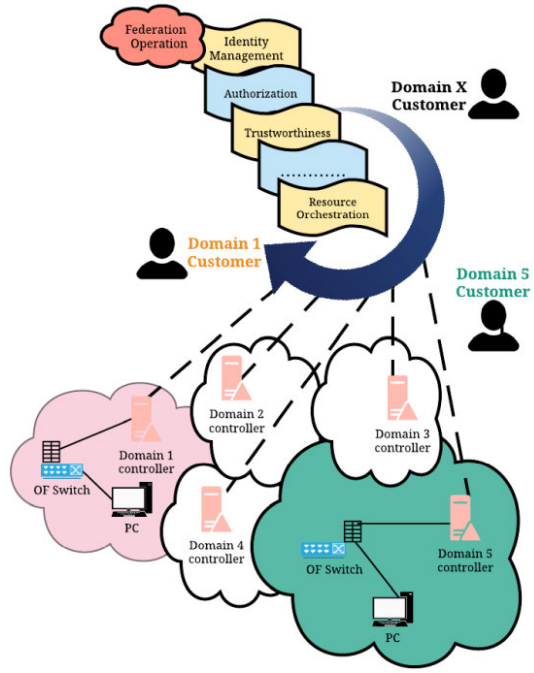

FIGURE 4. Simplified Federation Network.

While an AS is often referred to as a domain, a federation is a multi-domain system with physical or other relations between different domains as shown in Fig. 4. Operating and coordinating a multi-domain network can have challenges. To illustrate this, let's use the Internet as an example [110]. A network is built from a collection of AS that make independent decisions and exchange routing and reachability information by using a federation mechanism, in this case, BGP. AS provide routes to neighbouring AS, act as transit destinations and provide a ''cost'' for that route. Depending on the relative costs and peering agreements they have with other domains, AS can choose the most appropriate routes to use. AS have an implicit motivation to restrict traffic flows that do not match the AS policies and cost structure.

1) WHY FEDERATE?

The motivation for AS federation is to improve the operation of the Internet and to reduce operating costs. End-to-end delay, latency and packet loss are key factors for new and emerging near real time applications and services. The future operation of the Internet, enhancing quality of experience and reducing the cost of telecommunications are key drivers of the need for technology improvements, including AS federation. The notorious complexity of the current multi-domain routing system is difficult to operate and error-prone, resulting in multiple inefficiencies such as suboptimal inter-domain pathways and increased latency. Up to sixty per cent of all Internet paths today suffer from violations of triangular

inequality [111]. The current network ossification, which obstructs the introduction of new solutions, further aggravates the problem.

- Federating networks means sharing network resources among multiple independent but collaborative networks with the aim of optimising the use of these resources, improving the quality of network-based services and reducing service provision costs.
- SDN based federated networks will provide mutual benefits to the participating administrative authorities.

2) WHAT TO FEDERATE?

In a federated SDN based network the network devices and associated edge servers and other devices can be affected by the network control policies spanning more than one domain. Data collection, aggregation and submission to core data centres involving sensors and actuators that communicate with cyber-physical devices [112] and future vehicular networks are likely to be affected by factors the support federation. In a federated network environment, it is possible to identify the key actors as logical entities: resource consumers, service providers and trust providers [113] and their interests as bodies, actions and interfaces. Actions are applied to services to meet user demands for resource reservations. Network federation can involve different implementations and scenarios, e.g., a domain may connect with one or more neighbour domains utilising one or more optic fibres that are owned and operated by the AS operator or a third party. A dark fibre connection or wavelength-based transmission service could be employed. A transit link may be employed to provide a tunnel connection between AS and in this scenario the policies of the transit provider need to be considered when the domain federation is implemented. Infrastructure resources are combined to form elements of the federated network architecture [114]. organisations that operate globally may need to:

- deploy tiers of their applications across different time zones
- diversify their choice of cloud providers, for a number of reasons

A. BENEFITS OF FEDERATION

Multi-domain federation provides benefits including improved control, management and monitoring, cost savings, and improved end user quality of experience.

- Cost savings, achieved by resource sharing. Reduction of investment and leased circuit costs for the federation participating networks. Because these costs contribute to the overall cost of large-scale core networks.
- Enhanced multi-domain services, by unified management of the federation and improved service resilience. Service provisioning, in particular, can become easier and faster by simplifying the user perspective support structures. For large projects which are funded by several realms within the federation, this aspect is significant.
- Improved user experience. The interface between the federated network and its consumers must be abstract from the domains involved. It means users will not be aware of the advantages of the domain they are using. It makes it much easier to use the federated network from the user's point of view.

In an SDN based multi-domain federation users can benefit from improved on-demand provisioning of networking, storage and compute resources according to a pay-per-use business model [115].

B. FEDERATION USE CASES

SDN based programmable networks have been a significant part of current FI research. Researchers worldwide are interested in effective, consistent, practical, and reproducible environments that can be used to validate their proof-of-concept experiments and experiment with new algorithms, protocols, or network functions. This section provides use cases that can be used to:

- translate into requirements,
- provide a guide to describing the architecture,
- assist in the discovery of architectural elements, and
- to validate outcomes.

There are six different use scenarios [116], which are divided into two major clusters: network resources used to transfer data and the entire infrastructure is used to provide network-based services (including computing and storage resources). The clusters are respectively titled data domain and infrastructure domain. It is worth noting that use cases include both the data and infrastructure domains sharing similar design, trust and security assumptions [117].

1) DATA DOMAIN USE CASES

The data domain use cases focus on the efficient use of SDN technologies to provide linkages across geographically distributed networks with the opportunity to rapidly and efficiently realise data migration. The use cases include virtual infrastructure consisting of SDN islands interconnected with (inter-continental) dynamic circuit-switched networks (multi-domain transit networks). One essential aim is to optimise the use of inter-connectivity to realise data migration using SDN and Network Service Interface (NSI) operations between networks. The focus here is about coordinating caching, processing, and network services rather than the exact caching algorithms to be used, which are within the full range of user priorities and controls. Data must be moved from the origin to its destination endpoint usually on another SDN island. The following subsections describe each use case and explain how the data flows traverse networks.

Data-on-Demand. Distributed data delivery on-demand utilising managed data flows. This use case examines how vast volumes of data collected at diverse and distributed locations are handled. For example, several applications, such as astronomical observations or collaborative research, generate enormous amounts of data typically stored in dedicated

storage servers or devices in a nearby data centre. An application or user, i.e. a data processor, may want to run a post-processing algorithm on the data that various data providers collect. In this context, transferring the data from the original sites to the final location is neither necessary nor effective. An SDN based network that has a global view of the entire network might be more convenient and provide an intelligent solution as to when and how data might be moved across the network. The local controllers may create automatic links between different endpoints and guarantee end-to-end communication efficiency, with minimal delay and jitter.

Data pre-processing. The purpose of this use case is to provide near-real-time data, e.g. satellite images, to users located in different and remote locations without incurring the large Round Trip Delay (RTD) values typically found when using the public Internet. In this scenario, a dedicated platform could be strategically placed and the data pre-processed. This approach allocates resources for storage, pre-processing, caching and networking. It could also incorporate on-demand and application-driven network services for different data transfer needs which need well-defined network parameters. This approach [118] will also substantially reduce the amount of data to be transmitted through the transit network and increase the scenario efficiency.

Long distance high-quality media transmission. A rapid evolution of media content delivery is currently underway, especially in the context of ultra-high-definition of video streaming, i.e., 4K and 8K resolution. This development is directly related to demand for improved consumer experience and imposes higher bandwidth and lower network latency constraints. Hardware optimisation is required for data content transmission and reception, particularly in a very long-distance environment. At the same time, streamlining of the network is required in both the transport segments (NSI-enabled) and the inter-data centre networks (SDN-enabled). In this scenario, the deficiencies of legacy network control, management and monitoring can manifest in noticeable playback artefacts: jitter, incorrect frame spacing, interruption of transmission, etc. In addition, strict specifications are required to provide the consumer with ultra-high-definition and 3D video.

2) INFRASTRUCTURE DOMAIN USE CASES

Use cases for the infrastructure domain are primarily geared towards the use of a virtual distributed system that can be used to move entire workloads for data processing. Here, resources for networking, computation and storage are considered across the allocated physical space. This approach is in line with network service chaining and the ability to migrate network services, scale-out and scale-up facilities, as well as continuous service provision [119]. The rest of this section discusses use cases for the infrastructure domain and explains how the services can be implemented in a federated environment.

Inter-cloud use case. This use case focuses on the SDN technologies used with cloud systems and network infrastructure for a mission-critical, carrier-grade data mobility service. The distributed nature of complex cloud systems requires high throughput and reliable interconnects to transport data between the data centres. Consider, for example, a consumer heading to a distant location and to reduce latency and ensure that quality of service is maintained there is a need to migrate the user's profile and data to a data centre at the new location. This is a conventional mobility management requirement [120], however, depending on the particular user the workload associated with mobility could be considerable, and depending on the time of year, the number of mobility tasks could be significant.

Follow the sun (or moon). As detailed in [121], Internet usage curves follow a similar daily pattern globally and this means that there is a natural shift in the load on data centres during the day. The opposite is true during the night when data centre workloads involve backups, data migration, and other tasks that are minimised during peak times. This is often referred to as the ''follow the sun (or moon)'' principle. In addition, renewable energy prices are highly dependent on wind and solar energy (green energy) being available. In this scenario, one might move the load from one data centre to another using two different strategies: a) effectively moving the entire workload to a more efficient data centre by re-routing user traffic, or b) managing user requests at less efficient data centres by delegating the workflow to more efficient data centres. It is important to note that both scenarios involve end-to-end dynamic and on-demand connections between the federated data centres. However, a variety of different resources (network, compute and storage) need to be configured when the workload is transferred from one data centre to another.

Disaster recovery. In this use case data recovery via Infrastructure as a Service (IaaS) migration to a remote data centre occurs. Hardware and software solutions provide isolated tenants of physical resources (computers, storage, and network) in a multi-user data centre with a mechanism to ensure that data backups and recovery are available. Under specific conditions, such as a serious disaster, middleware can be used to migrate VMs and containers to a remote data centre to ensure business continuity.

3) WIRELESS NETWORKS

Wireless networks incorporate specific features like mobility management, dynamic channel configuration, and rapid client re-association. The value of SDN based wireless networking lies specifically in its ability to provide new capabilities, such as network slicing, and the creation of new services on top of the virtualised resources in the secure and isolated networks. In addition, one of the most important complementary technologies that can benefit by being SDN enabled is the smart home gateway. Service providers aim to provide home gateways with improved traffic control,

management and monitoring and with enhanced security. For example, [122] introduced a Virtual Home Gateway Control (VHC) for seamless mobility, service delivery and home energy management.

C. REQUIREMENTS FOR SDN FEDERATION

Domain federation can require an agreement between one or more AS administering organisations that will specify the terms under which the domain federation will occur, including management of the domain federation, security and what will happen in the event of equipment or system failure.

The following aspects need to be considered when design the federation agreement:

- Network topology. Routing and reachability is affected by network design and topology and what type of technologies are used.
- Network coverage. What will be included in the network federation agreement.
- Network utilisation. How the federated network connections will be used and resources allocated.
- Network design and upgrades. How the federated network will evolve over time and how changes might affect the federation.

The following problems may need to be addressed in the federation agreement:

1) **End-to-end connection.** The network federation framework needs to support multiple administrative domains. Federated networks may need to rely on dedicated inter-domain communication services that can be managed in deterministic (e.g. VPN, MPLS, SDWAN, dedicated leased lines, or optical circuits) methods based on connection-oriented paradigms and QoS [123].
2) **Standard interfaces and protocols.** Federation is facilitated when participating organzations use standard interfaces and protocols that manage the interconnection and control, management and monitoring information flow between gateways [124].
3) **SDN.** With the recent proliferation of SDN based technologies and solutions, it is likely that a federated network would be enhanced by the utilisation of SDN based technologies and solutions [125].
4) **Dynamic network resource allocation.** Aris et al. [102] suggests in order to enable aggregation, measurement and visualisation of applications to the distributed multi-domain federated network environment, different network assets need to be dynamically allocated according to meet the workflow requirements of deployed applications.
5) **Automated and intelligent service provisioning.** Federated networking scenarios create challenges related to specifications and critical features related to service delivery automation and resource management. SDN based network federation promotes enhanced multi-domain networks between enterprise and service provider data centres, helping clients access flexible bandwidth for ad-hoc applications, timely multi-domain workload migration and data processing [126].
6) **Vendor agnostic.** Network and data centre operators and enterprises have moved away from vendor lock-in to vendor agnostic networking. Federated networking in this scenario utilises standardised protocols and APIs to provide interoperability and to reduce operating costs [127].
7) **Access and transit technologies.** The selection of access and transit technologies can affect the operating cost and reliability of enterprise and data centre connectivity to gateways and other locations where network federation occurs [128].
8) **Resource accessibility.** Client access to resources in a federated networking environment should be transparent and automated utilising APIs or web services [129].
9) **Privacy and secure data management.** In a data centre scenario, VMs and containers should be logically isolated with networking implemented that, depending on identity and access credentials, provides authorised access, privacy and security. Federated networks require automatic trust-based access to resources and storage through a single protected identity management system [130].
10) **Network Function virtualisation.** NFV and logical network separation are utilised in enterprise and data centres to facilitate hosting systems and applications that have different access and resource demands [131]. Workload mobility, backups and load sharing federated network resources benefit when NFV is implemented.
11) **Policy (provisioning) framework.** Current research includes routing and reachability and to a lesser degree end-to-end policy management. This is because AS are controlled, managed and monitored by separate organisations that may or may not share the information needed to gain a global view of the network. It is a crucial challenge to orchestrate policy-based schemes into a federated network layer [132]. Federated networks require a new revolutionary architecture for multi-domain SDN with dynamic parameters.
12) **Orchestration of configuration tasks.** Existing Configuration & Management (C&M) techniques are based on languages with low-level procedural programming (general purpose or scripting). A controller can accept configuration demands or requests from multiple orchestrators in a federated environment. It is important to design end-to-end and automated (C&M) tasks in order to cope with the various resource requirements at different stages of the networking life-cycle [133].
13) **Management abstraction.** A network federation should comprise an abstraction layer that enables seamless access and management of external resources through various operating systems and virtualisation mechanisms [134].

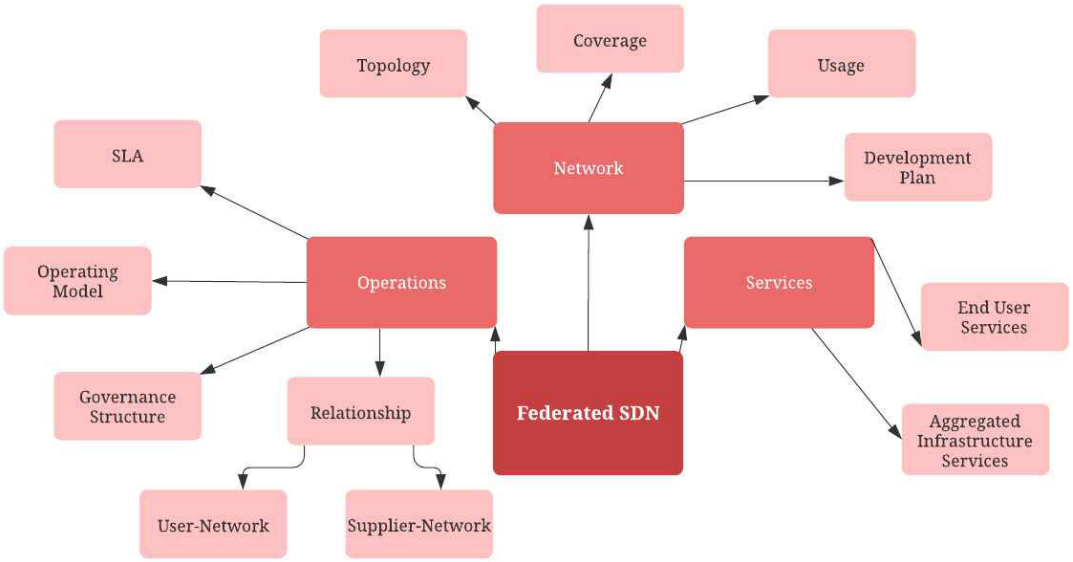

FIGURE 5. SDN based federation building blocks.

14) **Scalability, Flexibility, Agility.** Federated networking provides scalability, versatility, efficiency and cost reductions. To achieve scalability, flexibility and agility an SDN based federated network would maintain a global network view and optimise resource allocation [135].
15) **Guaranteed Network Performance.** Generally, variance in network load and performance, are symptoms of the ''best effort'' Internet. It is reasonable to assume that Quality of Service (QoS) specifications (e.g. minimum bandwidth) must be guaranteed for federated network interconnections in order to be able to control and manage traffic flows [136].

V. SDN BASED FEDERATED NETWORK BUILDING BLOCKS

The programmable control, management and monitoring of the infrastructure layer is a fundamental reason why SDN based federated networks are being deployed. A programmable forwarding plane enhances information dissemination, automatically path optimisation based on current traffic patterns, versatility, policies and management that are implemented using overlay applications. This section describes the three basic building blocks for SDN based federated networks: network, operations and services [137] as illustrated by Fig. 5.

- **Network**: The SDN enabled infrastructure layer is an important building block that provides the foundation upon which the programmability and intelligent applications can enhance the control, management and monitoring of the network and the traffic flows [138]. Management and allocation of the resources available in a network and between AS is a critical factor when the network operation is optimised to provide maximum performance. The increased complexity of the network and traffic flows necessitates improved near real-time management.
- **Operations**: Each federation member can manage operations and service delivery autonomously in a loosely coupled federation. Network operations for the networks of the federation partners, however, become directly interdependent in a tightly coupled federation [139]. Therefore, it is important to have an established set of defined procedures and workflows set out in an SLA, e.g., for fault handling and management. The federation member Network Operations Centres (NOC) implement and manage the processes and workflows. Operating requirements should be documented, understood by all stakeholders, and presented using an operating model that describes the relationship between the partners, the data sources used, and how traffic flows will be optimised. External SLAs can only be respected if partners implement internal procedures and practices to ensure that infrastructure and other resources are available and optimised. In a federated network scenario, relationships between partners can be both user-network and supplier-network relationships. A governance structure for the federated network operations should to be agreed upon, and a dispute resolution mechanisms established. The governance structure should be both efficient and fair; it could, for example, assign voting rights and agree on cost-sharing principles.
- **Services**: For the advantage of their end-users and transit networking partners, the federated network operators pool assets to develop, provide, and manage inter-network services. Services can be assembled utilising the federated network operators infrastructure, however, there could be a requirement to utilise third

party infrastructure and facilities, e.g., DCs and transit links. Network services play a dual role in a federated approach to networking: they are both end-user services, (e.g. an end-to-end Synchronous Optical Network -SONET) light path connecting an instrument to a computational site) and building blocks offered to the federated network by federation partners (e.g. a SONET connection between two federation partner point of presence). Choosing how to deliver services could result in quite different network architectures for the federation members. Service provision is no different in a core network, run as a single entity, from that of a typically built carrier or enterprise network. With this approach, on the level of service delivery, the federated network architecture is not directly visible [140]. On the other hand, with the aggregated infrastructure approach, service delivery itself becomes federated, offering services made up of network resources and systems. Therefore, this option is important for network design.

A. SDN BASED FEDERATION FRAMEWORK

Recognising that the SDN based federation is focused on agreed policies and practices, the underlying infrastructure should be available and optimised. An SDN based federation framework should include the capability to control, manage and monitor the resources available for the inter-network connections. The SDN controllers should be linked using east-west interfaces that provide standardised APIs. The application layer could be implemented utilising applications that maintain visibility of the federation member gateways, traffic flows and resources.

Unfortunately, existing SDN frameworks do not embrace multi-domain environments. Recently new research has commenced into SDN based multi-domain solutions. In Table 3 the provisioning (connection-oriented) and federated (slice-oriented) connection approaches [141] are compared.

- **The connection-oriented SDN based multi-domain framework**: focuses on creating circuits (connections) utilising overlay applications to manage the traffic flows between the federation member networks. Circuits can be set up in several ways – Layer 1 connections (port-port), Layer 2 streams, Layer 3 IP flows and Layer 4 TCP or UDP port connections. Management of legacy network connectivity add complexity to the inter-domain connectivity, but is it possible to manage the connectivity using an SDN based solution.
- **The slice-oriented SDN based multi-domain framework**: focuses on implementing distributed slices of networking resources as shown in Fig. 6. This approach utilises one logical domain to facilitate resource management. The systems that manage a slice-oriented federation include a link management service, a slicing management service (creating and handling slices) and a data centre management service to handle the storage and processing resources allocated to each slice. This strategy is based on an orchestrator that serves as a management platform and manages resources while also delegating the execution of operations to the domain infrastructure.

TABLE 3. Connection and slice oriented SDN based federation.

Connection-oriented	Slice-oriented
Providing circuits (connections) in the federated / multi-domain environment utilising SDN and OpenFlow protocol in particular	Focused on the definition of mechanisms in which distributed slices of networking resources can be created and used among many different SDN domains
OpenFlow enables circuits to be established in numerous forms	Enable the creation of sets of federated resources across several domains
SDN/OpenFlow capabilities in heterogeneous environments	Multi-domain federation entails (i) the SDN domains slices creation and (ii) E2E multi-domain connectivity among such slices to build the multi-domain federation
Technology agnostic's approach	Relies on an orchestrator that acts as management platform and coordinates services while yet delegating the execution of operations locally to each of the domain
Enables extension of the base functionalities without integration into core elements of the protocol	orchestrator can be seen as the entity that solves the gap between users (applications) and network, making use of several mechanisms (services)
Possibilities of the SDN/OpenFlow integration with the NSI- three possible solutions- based on STP definition, new service type and custom namespace	New architecture has been described

B. SDN BASED FEDERATION ARCHITECTURE

An SDN based federation architecture is a focus of current research. The SDN based federation architecture will be based on the policies and practices needed to control, manage and monitor the infrastructure used for the inter-network connections [142]. Following are the basic elements in the SDN based federation architecture:

- **Bodies**: Three types of logical bodies, resource users, resource suppliers, and trust providers, are typically involved with the federated network environment as outlined in Fig. 7.
- **Actions**: Actions are applied to resources to fulfil reservation requests based on resource type and policy. For example, network resources may expose isolated or shared access (e.g. optical cable, layer 2, or layer 3), normal or personalised packet switching, and different levels of service quality. Computing resources may expose physical systems, VMs or containers. Other resources may be utilised including instruments, sensors, and software services,e.g., firewalls, load-balancers, and encryption services.
- **Interfaces**: In a federated network environment the interfaces need to be standardised and secured. There is a requirement that the network operators implement monitoring and auditing to ensure the access to the interfaces is controlled. As there is a degree of trust

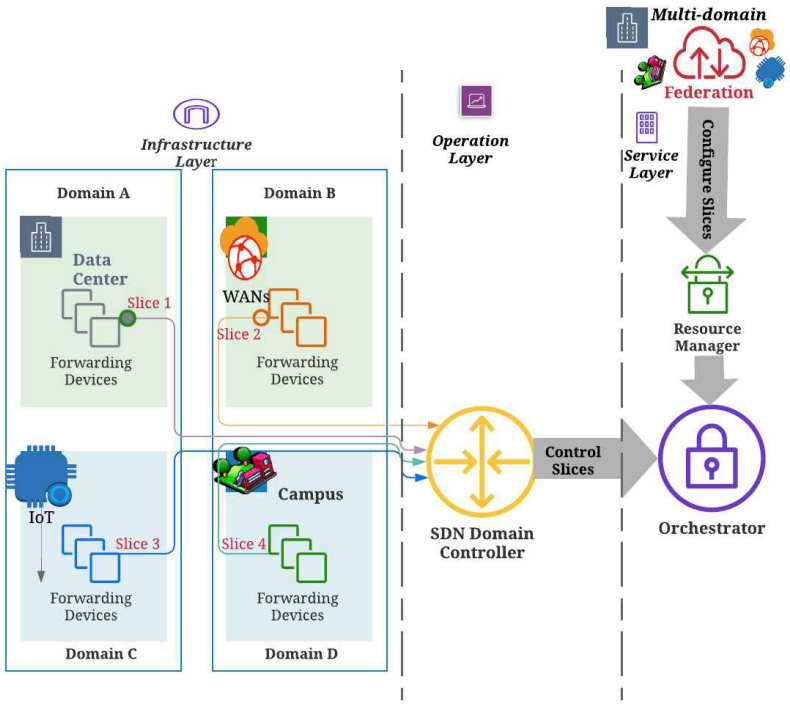

FIGURE 6. SDN based federation framework.

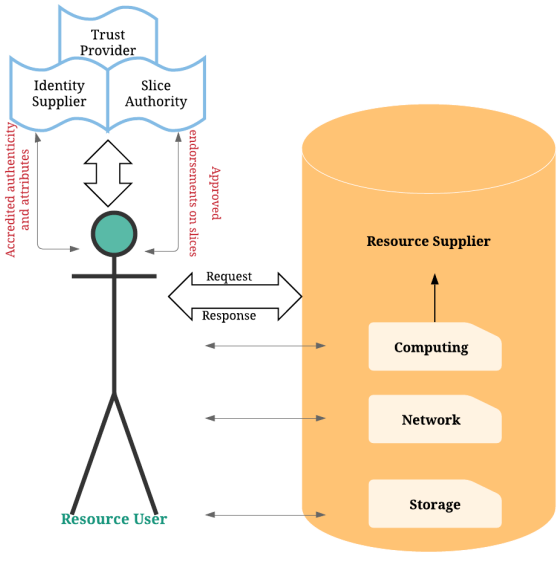

FIGURE 7. SDN based federation logical structure.

granted to the federation network members, it is important that coordinated access and authentication processes be implemented.

The development of an architecture that supports the underlying federation principle can be quantified by considering factors including:

- What is the right level of abstraction, the minimum collection of functionalities to be followed in order to share resources held by various authorities?
- What is the best-supporting governance model for subsidiarity? suppliers and the Facility itself that controls all of them.

Akhshabi and Dovrolis [143] took advantage of the internet architecture experience to create a model. It was based on two principles.

- The Internet ''Hourglass'' model defining the IP protocol as the convergence layer. For the Federation of Test bed Resources, one such convergence layer is required.
- The peering Internet model that depends on Customer Sand Providers and describes peering agreements in such a way as not to have a single control point. This helps to identify Experimenters, Test bed owners or suppliers and the Facility itself that controls all of them.

The following abstractions were defined:

- **Resource**: Management of the resources made available for the federated networking
- **User**: User identity guaranteed
- **Slice**: A distributed container in which resources are exchanged (sharing with VMs, in time, frequency, within flow space). It is also the basis for cost and resource accounting.

Slice Facility Architecture (SFA) was developed as an international effort to provide the minimum functionality necessary for a global federation trial to provide a stable shared API. As the understanding of the specifications progressed, the basic components for the federation trial were designed incrementally. In addition, a major effort has begun to supplement and fill the architecture with components that are

necessary for the whole life cycle of the SDN federation experiment. This includes the following steps [144]:

1) User account & slice development
2) Authentication
3) Resource exploration
4) Resource reservation & scheduling
5) Configuration & instrumentation
6) Execution
7) Results repatriation
8) Resource release

Step 1) is managed by the user's home authority, where the user has registered. Steps 2) to 4) and 8) are applicable to all of the authorities involved. Steps 5) to 8) are not in the SFA, but for this reason, other components such as Operating Management Framework (OMF) were created. OMF is a system for tracking, assessing and controlling.

SFA offers a secure API that allows authenticated and registered users to access the available resources and to delegate the resources that are required to conduct a particular experiment in compliance with the federation policies. SFA is used to federate the heterogeneous tools to be federated that belong to different administrative authorities. This will allow registered experimenters with these authorities to combine the available resources and perform advanced networking experiments involving wired and wireless technologies. The SFA layer includes the SFA Register, SFA Aggregate Managers (AMs) and drivers. It is the role of the SFA Registry to store the appropriate credentials for users and their slices.

C. UNIFIED DATA PLANE CONNECTIVITY

A homogenised connection method across the federated SDN based domains can be provided by the presence of unified architectures, such as SFA. However, this does not address the data plane stitching issue. A problem that federated design does not answer is how one virtual entity in one domain can interact with another virtual entity in a different domain at the data-link (Ethernet) layer. Techniques that enable the construction of virtual networks spanning resources are required for virtual resource utilisation in a networked environment. Typical network services found in a network environment, such as L2/L3 services, e.g. DNS and DHCP, should be provided by virtual networks with additional features, such as elasticity (up-scaling/down-scaling) and migration (rearrangement of resources and reconstruction of arbitrary network topologies over different physical infrastructure). With the presence of heterogeneous network resources that must be managed and configured within the federated environment, the issue of providing virtual networks becomes difficult. Argyropoulos et al. [145] presented a data plane stitching mechanism for federated heterogeneous virtual infrastructure. This technique operates in multi-tenant systems that require isolation, concurrency, programmability, and elasticity and is network virtualisation layer independent. As far as scalability is concerned, the heterogeneity of this mechanism does not impose restrictions on the overall number of concurrent network slices, nor does it result in any appreciable performance degradation in terms of end-to-end delay and bandwidth between virtual entities as compared to physical (substrate) entities. Additionally, the use of widely used network protocols (IEEE 802.1Q VLANs and GRE tunnels) ensures interoperability with a wide range of FI infrastructure.

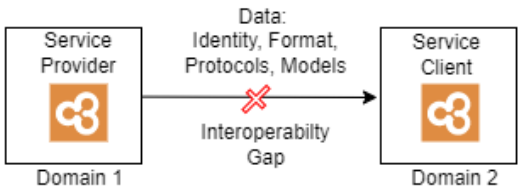

FIGURE 8. Data interoperability gap between heterogeneous domains.

In addition, the majority of the Internet today is heterogeneous, vendor-specific, vertically oriented, functionally fragmented, and locked behind closed identity and access management systems [146]. As shown in Fig. 8, it is not possible to create composite applications using services from different domains due to the interoperability gaps between the domains, entity interfaces, communication protocols, data formats, data models, and identity providers [147]. A recent effort in this area is Data Spine [148], a federated platform enabler that links and creates interoperability between domains. It offers the capability to federate the identity providers of various platforms, enables Single Sign On (SSO) functionality at the ecosystem level, supports interoperability between the services of various domains, and creates the technical foundation for quick, simple, and code-light development of cross-platform applications. From the perspectives of both service providers and service users, the methods for integrating synchronous (request-response) and asynchronous (publisher/subscriber) type services are also discussed in the Data Spine proposal.

VI. TEST BEDS

In research laboratories around the world, novel networking technologies are being created and demonstrated. The test beds developed and implemented by the researchers utilise software and hardware components and include monitoring and management systems. Research to address the requirements of the FI includes a new architecture that is based on a "clean slate." Federation extends infrastructure reach, both geographically and administratively. SDN federation has occurred across several dimensions for use by representatives of various research organisations. Although specifications and priorities vary, they all have to solve a similar set of problems. Such a federation suggests the fundamental needs of applications to interact across a generic, federated network. This section discusses a variety of current and planned SDN federation research efforts. A proposal for the federation of the test beds indicates that the research teams would be able to carry out distributed and collaborative experiments with FI

protocols and applications [149]. Table 4 provides a summary of the recent work on SDN based federation.

A. GENI

Global Ecosystem for Network Innovations (GENI) [150] offers a virtual laboratory for research and education into networking and distributed systems. GENI allows researchers to obtain computing resources from locations across the United States, connect computing resources using the 100 Gbps Internet2 Layer 2 service, install custom software or customised operating systems on the computing resources, control how network switches handle traffic flows in their experiments, and run applications at Layer 3 and above. GENI's architecture incorporates cloud federation. GENI administrators can determine the resources the community has at their disposal and thus a portion of GENI resources are reserved for internal use. The GENI architecture enables researchers to combine computing and storage resources. For full control of networking traffic, researchers may run their SDN controllers on a Layer 2 network domain. This capability is useful for the transfer of large volumes of scientific data.

B. OF@TEIN

At the Asia-Europe Summit (ASEM 3) in Seoul in 2000, the Trans-Eurasia Knowledge Network (TEIN) initiative was commenced. It was one of ASEM's most successful initiatives to connect ICT infrastructures between Asia and Europe. OpenFlow at TEIN (OF@TEIN) is a regional OpenFlow test bed project that began in 2012 with the cooperation of TEIN NRENs in South East Asia (SEA). It aims to promote SDN infrastructure via the TEIN network. The project [151] focuses on the following objectives to create a virtualised OpenFlow-enabled SDN test bed over international sites:

- SmartX rack design and checking
- OF@TEIN network deployment and test platform
- Development and deployment of SDN software at OF@TEIN

A unique SmartX rack was first developed for this project, with SDN based switching and remote management functions. The racks were installed at seven OF@TEIN sites (Vietnam, the Philippines, Thailand, Indonesia, Malaysia and two sites in Korea) and connected to allow for SDN experiments. Several SDN tools have been planned and built to support OF@TEIN test bed experiments and management. The project is currently looking forward to extending the OF@TEIN network by combining it with the organised creation of video-based medical cooperation and the global IPTV service.

C. FELIX

Federated test beds for Large-scale Infrastructure Experiments (FELIX) establish an integrated SDN test bed between Europe and Japan. It implements new APIs and logic for heterogeneous SDN and IT islands spread globally, enabling them to exchange information on resources, share overall resource pools, and provide dynamic network inter-connectivity between and within the islands [152]. Using the SDN control and NSI features to build SDN based federation services, FELIX offers new network management standards.

D. GÉANT

The GÉANT project's goal is to provide a pan-European network and related infrastructure for next generation networking research that will meet the connectivity needs of the affiliated research communities. The research needs include both a data-producing transport facility and a networking environment in which experiments can be carried out. At GÉANT, using SDN, the new "Test beds as a Service" [153] includes a dynamic packet network test bed service designed to tackle research on the upper network layers and dynamically construct a network consisting of a variety of network resources: routing, switching, computing, circuit and link, storage, and advanced experimental hardware or software.

E. OFELIA

Open Flow in Europe-Linking Infrastructure and Applications (OFELIA) is an experimental network designed to provide diverse OpenFlow-enabled infrastructure that allows experimentation with the SDN paradigm. OFELIA currently consists of ten sub-test beds (the so-called islands), most of which are in Europe and one in Brazil. In [154] a subset of the test bed resources, known as a slice, is made available including the programmable OpenFlow switches. Recently a new network virtualisation method named VeRTIGO was introduced to expand the way in which isolation is accomplished between slices, enabling arbitrary virtual network topologies on top of a physical test bed.

F. Fed4FIRE+

The Federation for Future Internet Research and Exploration (FIRE), Fed4FIRE (2012-2016), merged into Fed4FIRE+ (2017-2021), is an EU project aimed at developing a free, usable and secure platform for the cross-border federation of Internet research infrastructures. Here, sample community domains include optical and wireless networking, SDN, cloud and grid computing and smart cities. To ensure reliability and support of heterogeneous networks and experimentation requirements, heterogeneous communities, including the next-generation optical networking community, are involved in the project. The approach taken [155] includes:

- free tools for life-cycle control of experiments
- experimental measurements and monitoring
- trust and security processes
- inter-test bed enhanced communication capabilities

VII. OPPORTUNITIES AND CHALLENGES

A. OPPORTUNITIES

A list of SDN functionality that could be leveraged in federated environments is provided in [156]. As shown in Fig. 9,

TABLE 4. Summary of SDN federation test beds.

Test beds	Geographical Spread	Specifications	Prospects	Limitations
GENI [151]	Across United States	• Conducting networking and computer science research • Permitting evaluation of radical new network architectures including non-IP or even non-packet-based architectures	• Flexible and forward-looking, e.g., including processing, storage, and programmable network devices networked together by a variety of technologies • Include real-user traffic to permit realistic stressing	• There are technical and human interfaces which will need to be accommodated through the design and operation of the facility • Many aspects of the system are not defined such as authentication and authorisation interfaces
OF@TEIN [152]	Asia (Malaysia, Korea, Taiwan based) and Europe	• SDN-Cloud playground to interconnect distributed multi-domain cloud data centres • Software Defined Routing Exchange • Open stack cloud management software • Open Vswitch • ONOS controllers on utilising BGP as control plane • VLAN based multi-tenant traffic control (tagging, steering, mapping)	• Flexibility of networking control • Reasonable interconnections redundancy and performance enhancement • Easier to scale-out • End to end underlay aware interconnections with minimal configuration interference and performance impact on underlay infrastructure	• Low bandwidth throughput and connection instability, suboptimal routing • Hardware acceleration support for virtualisation and networking • Integrating several virtual switches is quite complex and challenging • Not large scale and routing redundancy configurations • Interoperability with other resources • Operational efficiency • Lack of automated DevOps style operation
FELIX [153]	Europe Japan funded	• Distributed multi-domain • Hierarchical model for inter-domain dependency management, with resource orchestrating entries • Cloud Stack • Open stack • SDN island, zone, slice • Resource orchestrators	• Resource reachability and coherent use of physical connections • Dynamic user-controlled construction of distributed virtual infrastructures • NSI-controlled transit domains • Users can request, monitor and manage a slice provisioned over distant SDN experimental facilities • Uses a combination of recursive and policy-based hierarchical configurations	• Resource orchestrating entities are limited for the synchronisation of resources available in particular administrative domains
GÉANT [154]	Pan-European network	• Test beds as a Service using SDN • The GN4 Phase 3 Network project (GN4-3N) is the most significant refresh of the GÉANT network in a decade	• Provides improved performance, increased flexibility, and reduced expense alongside long-term platform stability • Cost savings, support for multi-domain services and improved user experience	• The fault and performance management processes and tools being used in the domains have to be integrated to resolve issues • The whole federation needs to agree on common security • End-to-End Monitoring System
OFFELA [155]	Europe-Linking Infrastructure	• Allows researchers to not only experiment 'on' a test network but to control the network itself precisely and dynamically • Based on OpenFlow • Federated or interconnected islands	• High-performance OpenFlow equipment to ensure that the facility is based on mature technology • Allows to virtualise and control the network environment through secure and standardised interfaces	• Not scalable for large distributed architecture • Human intervention required • Requires authorisation and authentication policy
FED4FIRE+ [156]	EU project	• Focused on cloud computing- how to effectively offer different services without centralised or distributed control plane	• Testing measurements and monitoring • Trust and security processes • Inter-test bed enhanced communication capabilities	• Network connections between test beds are fixed and can't be manipulated using a dedicated system

the preliminary efforts offer a range of advantages as long as criteria are met. The federation of an SDN framework provides:

- rich and scalable traffic management capabilities
- a federated Internet that has improved operational efficiency, programmability and automated technology deployment capability
- new business models for operators and users, such as meeting the requirements of end-to-end QoS aware services

The list can be divided into two groups: firstly, the ability to dynamically build and change network services (automation), and secondly, improved interfaces that provide visibility

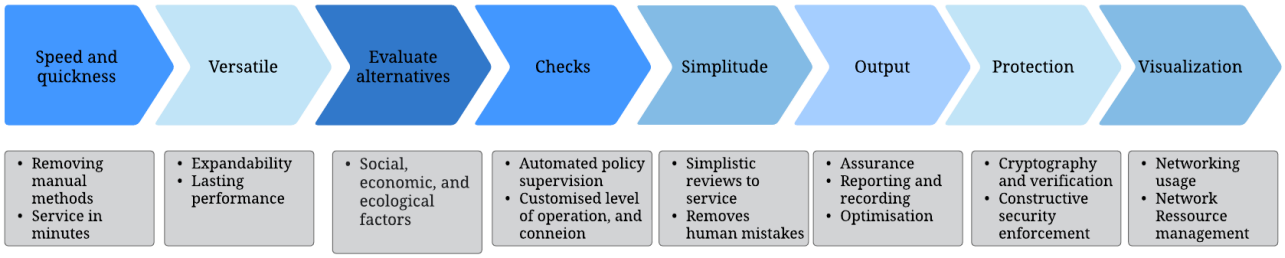

FIGURE 9. SDN federation opportunities.

(control). This section discusses the SDN based federation designs.

1) AUTOMATION, COMPATIBILITY AND INTEGRATION

Through automation, on-demand networks will respond to changes in services depending on the resources and specifications in place at the time. SDN based federation will also have greater interoperability and programmability [157]. A standard programming platform that can be used by interested parties simplifies orchestration and maintenance for operators, companies, independent developers, and users.

2) CONTROL, DISTINCTION, VISIBILITY, AGILITY

Since SDN has a standardised interface between the policy-controller and the lower-level flow devices, it provides the ability to enable, change, delete and transfer the domain resources. Policy-based controls monitor the bandwidth and service levels automatically, enabling consumers to configure specific facilities and access rights based on rules, regional areas and role-based access. SDN federation provides the ability to deliver unanticipated SDN services enabling enhanced network visibility with enhanced network situational awareness clarity and currency [158]. This in turn provides better quality of information about network services to make rapid and efficient decisions.

B. CHALLENGES

SDN based federation imposes demanding requirements and critical features in terms of service delivery automation, resource management or the development of on-demand network connectivity. Challenges and possible future directions are discussed in this section as outlined in Fig. 10.

1) SCALABILITY

A large number of network view requests can result in the control plane becoming overloaded. Routing entries from several domains can impose a strain on both the memory and the lookup capacity of the flow device forwarding engines. A challenge is to build a large multi-domain SDN network that avoids or solves the scalability problems [159].

2) FLEXIBILITY

SDN has the ability to enable the implementation and operation of more effective network applications and services and manage virtualised infrastructure. However, more recently SDN based networking research is tackling the flexibility needed for large network operation. The traditional centralised control mechanisms are difficult to utilise in federated networks and are not suited to the large-scale and increasing service requirements [160]. The SDN based networking control plane architecture is critical to integrated system flexibility. The modularity and flexibility of controller composition are still under investigation.

3) MIGRATION AND SUSTAINABLE DEPLOYMENT

The SDN based and conventional IP networks coexist using gateways. As SDN based networking expands into the remaining legacy networking environments there is a need to ensure that operators update existing systems and provide support for legacy equipment that may remain [161]. Challenges remain with the use of hybrid flow devices in SDN-enabled networks to support legacy systems [162]. Future research is needed to formulate strategies and interaction mechanisms that optimise the SDN based networking benefits while reducing the added complexity of paradigm coexistence.

4) SECURITY

Security is a key issue today and the deployment of new technologies that enhance security in a federated networking environment is important. In the heterogeneous networks, there is a critical need to provide enhanced privacy and protection. Compared to traditional networks, the highly programmable SDN based networking the potential of security threats to disrupt the network is far more serious [163]. Security research is ongoing and the multi-domain SDN security challenge remains.

5) RESILIENCY

In a multi-vendor, multi-tier service-based network the provision of enhanced resiliency is a challenge. Issues with a local device or service can create errors that can impact the entire network. There is a need for increased resiliency knowledge and management capability [164]. One of the key concerns when providing a virtualised network is what happens when a controller fails. Management of this situation typically revolves around a stable fallback configuration, enhanced redundancy and resiliency.

FIGURE 10. SDN federation challenges.

6) RELIABILITY

SDN controllers are used to control flow devices, to verify network topologies and to manage message flows in the event of equipment or system failure. In an SDN based network the controller is a critical element in the strategy to improve network reliability [165].

7) ROBUSTNESS

A functioning SDN based network should continue to work when the network degrades due to device or system failure or when traffic increases [166]. An important challenge is to design and implement systems that manage network degradation to provide robustness.

8) ORCHESTRATION

A challenge for SDN based networking is the provision of orchestration that facilitates solutions, applications and services from multiple vendors [167]. Orchestration has become complicated, with more and more features per end device. There is a growing need for SDN based service orchestration. An open controller northbound API that is standardised among vendors is required that can be used for consistent network management instructions that are deployed over the multi-vendor infrastructure.

9) COST MODEL

Resource sharing should result in cost-sharing for federated services and this a challenging problem [168]. Cost-sharing problems become more complex for on-demand services such as on-demand bandwidth. Costs are incurred for the on-demand services even though customer may not be fully utilising the resource and there is a financial risk of not being paid when offering on-demand services.

10) TECHNOLOGY DIFFERENCES AND A LACK OF STANDARDS

SDN controller and network management solution vendors are not focused on inter-operability in a multi-vendors infrastructure environment. The problem of technical discrepancies on the lower layers has to be resolved when networks are federated. Some domains are Synchronous Digital Hierarchy (SDH) based, others are MPLS based, or Ethernet. When federation happens there is a need to ensure that network and system interoperability is fully achieved [169].

11) PERFORMANCE EVALUATION

SDN based networking introduces performance evaluation challenges, particularly in a federated network environment. SDN performance questions have not been properly investigated [170]. Except for some commonly used time-consuming simulation and costly performance assessment experimental techniques, analytical modelling may permit the definition of a networking architecture that paves the way for network designers to establish performance outcomes. Despite evaluating the network architecture, there are other design aspects to be evaluated as well. Analytical models can include performance metrics related to routing, resource allocation algorithms and networking schemes. They can be used to capture network performance data, such as packet delivery rate, packet delay, jitter, buffer length, network throughput, and network blocking probability.

VIII. CONCLUSION

The introduction of SDN has succeeded in paving the way for next-generation networking. Consequently, multi-domain SDN based networking interconnection should become widespread. This paper discusses the benefits of the multi-domain SDN based network federation architecture and offers an understanding of its functionality. Due to ongoing research on the SDN model, some key problems influencing the architecture's effectiveness and full utilisation have been identified, focusing on scalability, flexibility, performance, security, and interoperability. The stakeholders involved in the SDN based network federation value chain face a range of barriers that need to be lowered to ensure that adoption occurs. A software-defined revolution would result from the optimisation of network administration made possible by the SDN based network federation paradigm, which is still being tested and developed across various platforms. It is anticipated that once the issues are dealt with, the SDN based network federation will ''sky-rocket'' to the pinnacle of technological development.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares. (Jan. 2006). *A Border Gateway Protocol 4 (BGP-4)*. RFC4271. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4271

[2] T. van Rossum, ''BGP security and the future: A meta-analysis of BGP threats and security to provide a new direction for practical BGP security,'' Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 2020.

[3] A. Abhashkumar, K. Subramanian, A. Andreyev, H. Kim, N. K. Salem, J. Yang, P. Lapukhov, A. Akella, and H. Zeng, "Running BGP in data centers at scale," in *Proc. NSDI*, 2021, pp. 65–81.

[4] J.-W. Hyun, K.-S. Yoo, D.-H. Hyun, and C.-K. Kim, "A study on the IoT LED streetlight convergence technology for smart city service," *J. Next-Gener. Converg. Technol. Assoc.*, vol. 4, no. 2, pp. 135–143, Apr. 2020.

[5] Open Networking Foundation. (Apr. 2014). *Software-Defined Networking: The New Norm for Networks*. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[6] G. Kulkarni, "Simplification of internet ossification through software defined network approach," *Global J. Comput. Sci. Technol.*, vol. 17, no. C3, pp. 25–29, Feb. 2018.

[7] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker, "Software-defined internet architecture: Decoupling architecture from infrastructure," in *Proc. 11th ACM Workshop Hot Topics Netw.*, Oct. 2012, pp. 43–48.

[8] E. Jacob, A. Mendiola, L. Podleski, R. Krzywania, M. Przywecki, K. Dombek, and A. Juszczyk, "Multi-domain software defined networking: Exploring possibilities," in *Proc. Sel. Papers Terena Netw. Conf.*, 2014, pp. 1–12.

[9] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "SeArch: A collaborative and intelligent NIDS architecture for SDN-based cloud IoT networks," *IEEE Access*, vol. 7, pp. 107678–107694, 2019.

[10] S. D. A. Shah, M. A. Gregory, S. Li, R. D. R. Fontes, and L. Hou, "SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13425–13442, Aug. 2022.

[11] P. Sermpezis and X. Dimitropoulos, "Can SDN accelerate BGP convergence? A performance analysis of inter-domain routing centralization," 2017, *arXiv:1702.00188*.

[12] J. Chen, X. Zheng, and C. Rong, "Survey on software-defined networking," in *Proc. 2nd Int. Conf. Cloud Comput. Big Data Asia*. Cham, Switzerland: Springer, 2015, pp. 115–124.

[13] 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, 2008.

[14] Z. Guo, Z. Guo, G. Shou, and Y. Hu, "An implementation of multi-domain software defined networking," in *Proc. 11th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, 2015, pp. 1–5.

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

[16] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, Jul. 2011.

[17] A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente, and D. Villela, "A survey of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 2, pp. 7–23, Apr. 1999.

[18] L. Zhang, "Named data networking (NDN) project," Xerox Palo Alto Research Center, Palo Alto, CA, USA, Tech. Rep., NDN-0001, 2010, p. 158.

[19] L. Popa, A. Ghodsi, and I. Stoica, "HTTP as the narrow of the future internet," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, Oct. 2010, pp. 1–6.

[20] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[21] A. Moreno, B. Curto, and V. Moreno, "New protocol for grouping data using active network," in *Network Control and Engineering for QoS, Security and Mobility, III*, D. Gaïti, S. Galmés, and R. Puigjaner, Eds. Boston, MA, USA: Springer, 2005, pp. 205–217.

[22] D. S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith, "The switchware active network architecture," *IEEE Netw.*, vol. 12, no. 3, pp. 29–36, Jun. 1998.

[23] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek, "The click modular router," *Operating Syst. Rev.*, vol. 18, no. 3, pp. 263–297, 1999.

[24] M. Handley, O. Hodson, and E. Kohler, "XORP: An open platform for network research," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 53–57, Jan. 2003.

[25] P. Jakma. *Quagga Software Routing Suite*. [Online]. Available: https://www.nongnu.org/quagga/docs.html

[26] CZ. NIC Labs. *The BIRD Internet Routing Daemon*. [Online]. Available: https://bird.network.cz/

[27] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proc. ACM SIGCOMM Workshop Future Directions Netw. Archit.*, Aug. 2004, pp. 5–12.

[28] L. Yang, R. Dantu, T. Anderson, and R. Gopal, "Forwarding and control element separation (ForCES)," Tech. Rep., RFC3746, Apr. 2004. [Online]. Available: https://www.hjp.at/doc/rfc/rfc3746.html

[29] "The soft router architecture," in *Proc. Workshop Hot Topics Netw. (ACM SIGCOMM)*, 2004, pp. 1–6.

[30] W. Wang, L. Dong, B. Zhuge, M. Gao, F. Jia, R. Jin, J. Yu, and X. Wu, "Design and implementation of an open programmable router compliant to IETF ForCES specifications," in *Proc. 6th Int. Conf. Netw. (ICN)*, Apr. 2007, p. 82.

[31] A. Doria, J. H. Salim, R. Haas, H. M. Khosravi, W. Wang, L. Dong, R. Gopal, and M. J. Halpern, "Forwarding and control element separation (ForCES) protocol specification," Tech. Rep., RFC5810, Mar. 2010.

[32] J. Rexford, A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, G. Xie, J. Zhan, and H. Zhang, "Network-wide decision making: Toward a wafer-thin control plane," in *Proc. HotNets*, 2004, pp. 59–64.

[33] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, 2005.

[34] H. Yan, D. A. Maltz, T. E. Ng, H. Gogineni, H. Zhang, and Z. Cai, "Tesseract: A 4D network control plane," in *Proc. NSDI*, vol. 7, 2007, p. 27.

[35] J. Vasseur, A. Farrel, and G. Ash. (Aug. 2006). *A Path Computation Element (PCE)-Based Architecture*. RFC. [Online]. Available: https://rfc-editor.org/rfc/rfc4655.txt

[36] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "SANE: A protection architecture for enterprise networks," in *Proc. USENIX Secur. Symp.*, vol. 49, 2006, p. 50.

[37] J. Luo, J. Pettit, M. Casado, J. Lockwood, and N. McKeown, "Prototyping fast, simple, secure switches for etha," in *Proc. 15th Annu. IEEE Symp. High-Perform. Interconnects*, Aug. 2007, pp. 73–82.

[38] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.

[39] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1270–1283, Aug. 2009.

[40] C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep *Q*-network," in *Proc. Int. Conf. Comput. Data Social Netw.*, vol. 12575. Dallas, TX, USA: Springer, Dec. 2020, p. 26.

[41] (Dec. 2020). *Open Networking Foundation*. [Online]. Available: https://opennetworking.org/sdn-definition/

[42] F. A. Lopes, M. Santos, R. Fidalgo, and S. Fernandes, "A software engineering perspective on SDN programmability," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1255–1272, 2nd Quart., 2016.

[43] P. Newman, G. Minshall, and T. L. Lyon, "IP switching-ATM under IP," *IEEE/ACM Trans. Netw.*, vol. 6, no. 2, pp. 117–129, Apr. 1998.

[44] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.

[45] H. Jamjoom, D. Williams, and U. Sharma, "Don't call them middleboxes, call them middlepipes," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, Aug. 2014, pp. 19–24.

[46] A. Gember, P. Prabhu, Z. Ghadiyali, and A. Akella, "Toward software-defined middlebox networking," in *Proc. 11th ACM Workshop Hot Topics Netw.*, Oct. 2012, pp. 7–12.

[47] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.

[48] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, 2010, vol. 10, no. 8, pp. 89–92.

[49] M. Ghobadi, S. H. Yeganeh, and Y. Ganjali, "Rethinking end-to-end congestion control in software-defined networks," in *Proc. 11th ACM Workshop Hot Topics Netw.*, Oct. 2012, pp. 61–66.

[50] N. Handigol, M. Flajslik, S. Seetharaman, N. McKeown, and R. Johari, "Aster*x: Load-balancing as a network primitive," in *Proc. 9th GENI Eng. Conf.*, 2010, pp. 1–2.

[51] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks," in *Proc. NSDI*, vol. 10, 2010, pp. 249–264.

[52] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Participatory networking: An API for application control of SDNs," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, New York, NY, USA, Aug. 2013, pp. 327–338.

[53] K. Jeong, J. Kim, and Y.-T. Kim, "QoS-aware network operating system for software defined networking with generalized OpenFlows," in *Proc. IEEE Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 1167–1174.

[54] S. Ong, "Network transformation with software-defined networking and Ethernet fabrics," Brocade Communications Systems, San Jose, CA, USA, Tech. Rep., 2012.

[55] H. Alkhatib, P. Faraboschi, E. Frachtenberg, H. Kasahara, D. Lange, P. Laplante, A. Merchant, D. Milojicic, and K. Schwan, "IEEE CS 2022 report," IEEE Comput. Soc., Washington, DC, USA, Tech. Rep., 2014.

[56] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control plane of software defined networks: A survey," *Comput. Commun.*, vol. 67, pp. 1–10, Aug. 2015.

[57] D. Erickson, "The beacon openflow controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 13–18.

[58] A. Voellmy and J. Wang, "Scalable software defined network controllers," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2012, pp. 289–290.

[59] Z. Cai, A. L. Cox, and T. Ng, "Maestro: A system for scalable OpenFlow control," Rice Univ., Houston, TX, USA, Tech. Rep., 2010.

[60] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Oct. 2010.

[61] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, 2011, pp. 254–265.

[62] X. Gao, L. Kong, W. Li, W. Liang, Y. Chen, and G. Chen, "Traffic load balancing schemes for devolved controllers in mega data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 572–585, Feb. 2017.

[63] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, and R. Ramanathan, "Onix: A distributed control platform for large-scale production networks," in *Proc. OSDI*, vol. 10, 2010, pp. 1–6.

[64] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, vol. 3, 2010, pp. 1–6.

[65] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–4.

[66] A. Singla and B. Rijsman. (Nov. 2013). *Day One: Understanding Opencontrail Architecture*. Juniper Network. [Online]. Available: https://www.juniper.net/documentation/en_US/day-one-books/OpenContrailBook.pdf

[67] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 19–24.

[68] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized? State distribution trade-offs in software defined networks," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 1–6.

[69] A. S.-W. Tam, K. Xi, and H. J. Chao, "Use of devolved controllers in data center networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2011, pp. 596–601.

[70] S. Schmid and J. Suomela, "Exploiting locality in distributed SDN control," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 121–126.

[71] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 7–12.

[72] A. Krishnamurthy, S. P. Chandrabose, and A. Gember-Jacobson, "Pratyaastha: An efficient elastic distributed SDN control plane," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, Aug. 2014, pp. 133–138.

[73] F. Le, C. Leet, C. Makaya, M. Rio, X. Wangp, and Y. R. Yang, "SFP: Toward a scalable, efficient, stable protocol for federation of software defined networks," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov.*, Aug. 2017, pp. 1–6.

[74] Y. Zhang, B. Zhu, Y. Fang, S. Guo, A. Zhang, and S. Zhong, "Secure inter-domain forwarding loop test in software defined networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 162–178, Jan. 2020.

[75] C. J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 198–211, Jul. 2013.

[76] M. Bonola, G. Bianchi, G. Picierro, S. Pontarelli, and M. Monaci, "StreaMon: A data-plane programming abstraction for software-defined stream monitoring," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 6, pp. 664–678, Nov./Dec. 2017.

[77] M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. W. Hong, "A flow-based method for abnormal network traffic detection," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, vol. 1, Apr. 2004, pp. 599–612.

[78] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 151–152.

[79] H. Cui, G. O. Karame, F. Klaedtke, and R. Bifulco, "On the fingerprinting of software-defined networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 10, pp. 2160–2173, May 2016.

[80] Z. Wu, L. Zhang, and M. Yue, "Low-rate DoS attacks detection based on network multifractal," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 5, pp. 559–567, Sep. 2015.

[81] A. Merlo, M. Migliardi, N. Gobbo, F. Palmieri, and A. Castiglione, "A denial of service attack to UMTS networks using SIM-less devices," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 3, pp. 280–291, May/Jun. 2014.

[82] E. Tantar, M. Palattella, T. Avanesov, M. Kantor, and T. Engel, "Cognition: A tool for reinforcing security in software defined networks," in *Advances in Intelligent Systems and Computing*, vol. 288, 2014, pp. 61–78.

[83] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "OrchSec: An orchestrator-based architecture for enhancing network-security using network monitoring and SDN control functions," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[84] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 121–126.

[85] T. Hinrichs, N. Gude, M. Casado, J. Mitchell, and S. Shenker, "Expressing and enforcing flow-based network security policies," Univ. Chicago, Chicago, IL, USA, Tech. Rep., 9, 2008.

[86] S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model checking invariant security properties in OpenFlow," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 1974–1979.

[87] N. Handigol, B. Heller, V. Jeyakumar, D. Maziéres, and N. McKeown, "Where is the debugger for my software-defined network?" in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 55–60.

[88] S. Matsumoto, S. Hitz, and A. Perrig, "Fleet: Defending SDNs from malicious administrators," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, Aug. 2014, pp. 103–108.

[89] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for openflow applications," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 171–172.

[90] C. Peng, Q. Zhang, and C. Tang, "Improved TLS handshake protocols using identity-based cryptography," in *Proc. Int. Symp. Inf. Eng. Electron. Commerce*, May 2009, pp. 135–139.

[91] H. E. Egilmez and A. M. Tekalp, "Distributed QoS architectures for multimedia streaming over software defined networks," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1597–1609, Oct. 2014.

[92] F. X. A. Wibowo, M. A. Gregory, K. Ahmed, and K. M. Gomez, "Multi-domain software defined networking: Research status and challenges," *J. Netw. Comput. Appl.*, vol. 87, pp. 32–45, Jun. 2017.

[93] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of

your production networks with OpenFlow," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 129–130, Jan. 2010.

[94] 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," in *Proc. ACM SIGCOMM Conf.*, Aug. 2013, pp. 475–476.

[95] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A software defined internet exchange," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 551–562, 2014.

[96] P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, and Y. Lin, "A west-east bridge based SDN inter-domain testbed," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 190–197, Feb. 2015.

[97] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, and R. Sidi, "SDNi: A message exchange protocol for software defined networks (SDNs) across multiple domains," Tech. Rep., 2012.

[98] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, and S. Venkata, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[99] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven WAN," in *Proc. ACM SIGCOMM Conf.*, Aug. 2013, pp. 15–26.

[100] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Comput. Netw.*, vol. 75, pp. 453–471, Dec. 2014.

[101] J. Chung, H. Owen, and R. Clark, "SDX architectures: A qualitative analysis," in *Proc. SoutheastCon*, Mar. 2016, pp. 1–8.

[102] A. C. Risdianto, P.-W. Tsai, T. C. Ling, C.-S. Yang, and J. Kim, "Enhanced ONOS SDN controllers deployment for federated multi-domain SDN-cloud with SD-routing-exchange," *Malaysian J. Comput. Sci.*, vol. 30, no. 2, pp. 134–153, Jun. 2017.

[103] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, and B. Lantz, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

[104] H. Zhou, C. Wu, Q. Cheng, and Q. Liu, "SDN-LIRU: A lossless and seamless method for SDN inter-domain route updates," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2473–2483, Aug. 2017.

[105] J. Wang, G. Shou, Y. Hu, and Z. Guo, "A multi-domain SDN scalability architecture implementation based on the coordinate controller," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2016, pp. 494–499.

[106] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[107] D. Kim and Y.-H. Kim, "A network federation scheme for inter-domain SDN communications," *Webology*, vol. 19, no. 1, pp. 4515–4526, Jan. 2022.

[108] (Dec. 2020). [Online]. Available: https://opennetworking.org/https://www.geni.net/

[109] (Mar. 2014). [Online]. Available: https://www.nuagenetworks.net/blog/federation-promises/

[110] J. Aznar, E. Escalona, I. Canyameres, O. Moya, and A. Vines, "CNSMO: A network services manager/orchestrator tool for cloud federated environments," in *Proc. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2016, pp. 1–5.

[111] V. Kotronis, X. Dimitropoulos, R. Klöti, B. Ager, P. Georgopoulos, S. Schmid, M. Yu, and R. Fonseca, "Control exchange points: Providing QoS-enabled end-to-end services via SDN-based inter-domain routing orchestration," in *Proc. Open Netw. Summit*, 2014, pp. 1–15.

[112] K.-C. Wang, M. Brinn, and J. Mambretti, "From federated software defined infrastructure to future internet architecture," in *Proc. 1st Int. Sci. Technol. Conf.*, Oct. 2014, pp. 1–6.

[113] (Jan. 2021). *Scaling Trustworthy Access to Global Research and Collaboration*. [Online]. Available: https://www.incommon.org/federation/

[114] B. Belter, M. Cosin, P. van Daalen, L. Fischer, I. Golub, A. Hanemann, M. Kaat, M. Przywecki, B. Radojevic, S. Tyley, and S. Vukovojac, "Deliverable DJ1.3.1: Architecture considerations for federated backbone networks study," Tech. Rep., Oct. 2010.

[115] S. Azodolmolky, P. Wieder, and R. Yahyapour, "SDN-based cloud computing networking," in *Proc. 15th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jun. 2013, pp. 1–4.

[116] G. Carrozzo, R. Monno, B. Belter, R. Krzywania, K. Pentikousis, M. Broadbent, T. Kudoh, A. Takefusa, A. Vieo-Oton, C. Fernandez, B. Puvpe, and J. Tanaka, "Large-scale SDN experiments in federated environments," in *Proc. Int. Conf. Smart Commun. Netw. Technol. (SaCoNeT)*, Jun. 2014, pp. 1–6.

[117] D. S. Alexander, W. A. Arbaugh, A. D. Keromytis, and J. M. Smith, "A secure active network environment architecture: Realization in SwitchWare," *IEEE Netw.*, vol. 12, no. 3, pp. 37–45, May 1998.

[118] C. Fernandez, C. Bermudo, G. Carrozzo, R. Monno, B. Belter, K. Pentikousis, U. Toseef, T. Kudoh, A. Takefusa, J. Haga, B. Puype, and J. Tanaka, "A recursive orchestration and control framework for large-scale, federated SDN experiments: The FELIX architecture and use cases," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 30, no. 6, pp. 428–446, Nov. 2015.

[119] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *Proc. IEEE SDN Future Netw. Services*, Nov. 2013, pp. 1–7.

[120] K. Pentikousis and P. Bertin, "Mobility management in infrastructure networks," *IEEE Internet Comput.*, vol. 17, no. 5, pp. 74–79, Sep. 2013.

[121] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. ACM SIGCOMM Conf. Data Commun.*, Aug. 2009, pp. 123–134.

[122] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: Empowering research in mobile networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125–126, 2010.

[123] I. Petri, M. Zou, A. R. Zamani, J. Diaz-Montes, O. Rana, and M. Parashar, "Integrating software defined networks within a cloud federation," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 179–188.

[124] J. Sen, "Security and security and privacy privacy privacy issues in cloud computing computing," Innov. Labs, Tata Consultancy Service Ltd., Kolkata, India, Tech. Rep., 2019.

[125] F. X. A. Wibowo and M. A. Gregory, "Software defined networking properties in multi-domain networks," in *Proc. 26th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Dec. 2016, pp. 95–100.

[126] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619304773

[127] J. Opara-Martins, R. Sahandi, and F. Tian, "Critical analysis of vendor lock-in and its impact on cloud computing migration: A business perspective," *J. Cloud Comput.*, vol. 5, no. 1, pp. 1–18, Dec. 2016.

[128] Y. Wang and J. Bi, "Survey of mechanisms for inter-domain SDN," *ZTE Commun.*, vol. 15, no. 3, p. 1, 2017.

[129] H. Mostafaei, G. Lospoto, A. Brandimartey, R. Di Lallo, M. Rimondini, and G. D. Battista, "SDNS: Exploiting SDN and the DNS to exchange traffic in a federated network," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–5.

[130] D. Kidston and T. Willink, "Investigations in software-defined networking (SDN)," Communications Research Centre, Ottawa, ON, Canada, Tech. Rep., 2.1, Feb. 2016.

[131] A. Levin and P. Massonet, "Enabling federated cloud networking," in *Proc. 8th ACM Int. Syst. Storage Conf.*, May 2015, p. 1.

[132] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, and D. Hutchison, "Network service orchestration standardization: A technology survey," *Comput. Standards Interfaces*, vol. 54, pp. 203–215, Jan. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548916302458

[133] (Oct. 2016). [Online]. Available: http://datavision-Inc.com/wp-content/uploads/2020/09/Datavision-White-Paper-SDN-Federation.pdf

[134] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "FlowVisor: A network virtualization layer," *OpenFlow Switch Consortium*, vol. 1, p. 132, Oct. 2009.

[135] M. Liyanage, A. Gurtov, and M. Ylianttila, *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*. Hoboken, NJ, USA: Wiley, 2015.

[136] F. Alharbi, "SDN-based mechanisms for provisioning quality of service to selected network flows," Ph.D. dissertation, Univ. Kentucky, Kentucky, U.K., 2018.

[137] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," 2014, *arXiv:1406.0440*.

[138] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *Queue*, vol. 11, no. 12, pp. 20–40, Dec. 2013.

[139] K. Feeney, R. Brennan, J. Keeney, H. Thomas, D. Lewis, A. Boran, and D. O'Sullivan, "Enabling decentralised management through federation," *Comput. Netw.*, vol. 54, no. 16, pp. 2825–2839, Nov. 2010.

[140] P. Bhoj, S. Singhal, and S. Chutani, "SLA management in federated environments," *Comput. Netw.*, vol. 35, no. 1, pp. 5–24, Jan. 2001.

[141] R. Zahradnícek, "Multi-domain SDN tools for creation of virtual private networks," M.S. thesis, 2017.

[142] S. Fdida, T. Korakis, H. Niavis, S. Salsano, and G. Siracusano, "The EXPRESS SDN experiment in the OpenLab large scale shared experimental facility," in *Proc. 1st Int. Sci. Technol. Conf.*, Oct. 2014, pp. 1–7.

[143] S. Akhshabi and C. Dovrolis, "The evolution of layered protocol stacks leads to an hourglass-shaped architecture," in *Proc. ACM SIGCOMM Conf.*, New York, NY, USA, 2011, pp. 206–217.

[144] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "OMF: A control and management framework for networking testbeds," *ACM SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, pp. 54–59, Jan. 2010.

[145] C. Argyropoulos, G. Androulidakis, D. Kalogeras, B. Pietrzak, B. Belter, L. Lymberopoulos, and V. Maglaris, "Network virtualization over heterogeneous federated infrastructures: Data plane connectivity," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, May 2013, pp. 26–33.

[146] A. Singh, G. Ormazabal, H. Schulzrinne, Y. Zou, P. Thermos, and S. Addepalli, "Unified heterogeneous networking design," in *Principles, Systems and Applications of IP Telecommunications*, Oct. 2013, pp. 1–7.

[147] T. Tarui, Y. Kanada, M. Hayashi, and A. Nakao, "Federating heterogeneous network virtualization platforms by slice exchange point," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 746–749.

[148] R. A. Deshmukh, D. Jayakody, A. Schneider, and V. Damjanovic-Behrendt, "Data spine: A federated interoperability enabler for heterogeneous IoT platform ecosystems," *Sensors*, vol. 21, no. 12, p. 4010, Jun. 2021.

[149] D. Larrabeiti, L. Kazovsky, G. Rodriguez, R. Aparicio, T. S. Shen, and S. Yin, "Integrating a next-generation optical access network testbed into a large-scale virtual research testbed," in *Proc. 17th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2015, pp. 1–6.

[150] T. Hwang, "NSF GENI cloud enabled architecture for distributed scientific computing," in *Proc. IEEE Aerosp. Conf.*, Mar. 2017, pp. 1–8.

[151] P.-W. Tsai, A. C. Risdianto, T. C. Ling, J. Kim, and C.-S. Yang, "Design and implementation of monitoring schemes for software-defined routing over a federated multi-domain SDN testbed," in *Proc. Asia–Pacific Adv. Netw.*, vol. 42, 2016, pp. 27–33.

[152] U. Toseef, C. Fernandez, C. Bermudo, G. Carrozzo, R. Monno, B. Belter, K. Dombek, L. Ogrodowczyk, T. Kudoh, A. Takefusa, J. Haga, T. Ikeda, J. Tanaka, and K. Pentikousis, "Implementation of the FELIX SDN experimental facility," in *Proc. 4th Eur. Workshop Softw. Defined Netw.*, Sep. 2015, pp. 7–12.

[153] L. Fischer, B. Belter, M. Przywecki, M. Cosin, P. Van Daalen, M. Kaat, I. Golub, B. Radojevic, S. Vukovojac, and A. Hanemann, "Building federated research networks in Europe," in *Proc. Terena Netw. Conf.*, 2010, pp. 1–13.

[154] M. Gerola, R. D. Corin, R. R. F. De Pellegrini, E. Salvadori, H. Woesner, T. Rothe, M. Sune, and L. Bergesio, "Demonstrating inter-testbed network virtualization in OFELIA SDN experimental facility," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2013, pp. 39–40.

[155] B. Devi, "SDN/NFV for flexible federation of SATOPS networks," in *Proc. Group Syst. Archit. Workshop*, 2017, pp. 1–21.

[156] Y. Wang and I. Matta, "Multi-layer virtual transport network management," *Comput. Commun.*, vol. 130, pp. 38–49, Oct. 2018.

[157] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A survey on data plane flexibility and programmability in software-defined networking," *IEEE Access*, vol. 7, pp. 47804–47840, 2019.

[158] J. Santos, "Scalable design of SDN controllers for optical networks using federation-based architectures," in *Proc. 21st Eur. Conf. Netw. Opt. Commun. (NoC)*, Jun. 2016, pp. 70–75.

[159] M.-Y. Luo and J.-Y. Chen, "Software defined networking across distributed datacenters over cloud," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, vol. 1, Dec. 2013, pp. 615–622.

[160] L. Bai, T. de Cola, Q. Yu, and W. Zhang, "Space information networks," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 8–9, Jan. 2019.

[161] T. Benson, A. Akella, and D. A. Maltz, "Unraveling the complexity of network management," in *Proc. NSDI*, 2009, pp. 335–348.

[162] K. Giotis, G. Androulidakis, and V. Maglaris, "Leveraging SDN for efficient anomaly detection and mitigation on legacy networks," in *Proc. 3rd Eur. Workshop Softw. Defined Netw.*, Sep. 2014, pp. 85–90.

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

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

[165] A. Metzler and A. Metzler, "Ten things to look for in an SDN controller," Tech. Rep., 2013.

[166] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," Tech. Rep., 1999.

[167] F. Risso. (Dec. 2020). *Orchestration in a Real Network: A Case Study*. [Online]. Available: https://sdn.ieee.org/images/files/pdf/FederatedTestbeds/may2016_risso_-_orchestration-in-a-real-network-a-case-study.pdf

[168] J. Famaey, S. Latre, T. Wauters, and F. De Turck, "FedRR—A federated resource reservation algorithm for multimedia services," in *Proc. IEEE Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 130–137.

[169] C. Gaul, M. Korner, and O. Kao, "Design and implementation of a cloud-federation agent for software defined networking," in *Proc. IEEE Int. Conf. Cloud Eng.*, Mar. 2015, pp. 323–328.

[170] K. Kondepu, A. Sgambelluri, F. Cugini, P. Castoldi, R. A. Morenilla, D. Larrabeiti, B. Vermeulen, and L. Valcarenghi, "Performance evaluation of SDN-controlled green mobile fronthaul using a federation of experimental network testbeds," *Photonic Netw. Commun.*, vol. 37, no. 3, pp. 399–408, Jun. 2019.

MOHAMMAD HASSAN received the Graduate Diploma degree in engineering management, Certificate II and III in telecommunications, the B.Sc. degree (Hons.) in electronics and computer science, and the M.Eng. degree in telecommunications engineering. He is currently coordinating computer systems and network engineering for the Advanced Diploma and Associate Degree Programs in the future technologies industry cluster at the College of Vocational Education, RMIT University. He is also a Cisco Certified Academy Instructor (CCAI). He has 15 years of teaching experience. He is a member of the Australia's Victorian ElectroComms and the ICT Senate Committee.

MARK A. GREGORY (Senior Member, IEEE) was born in Melbourne, Australia. He received the B.Eng. degree (Hons.) in electrical engineering from the University of New South Wales, Sydney, Australia, in 1984, and the M.Eng. and Ph.D. degrees from RMIT University, Melbourne, Australia, in 1992 and 2008, respectively. He is currently an Associate Professor with the School of Engineering, RMIT University.

SHUO LI (Member, IEEE) received the B.Eng. and Ph.D. degrees from the City University of Hong Kong, Hong Kong, in 2009 and 2014, respectively. She is currently a Lecturer with the School of Engineering, RMIT University, Australia. Her research interests include analysis and design of telecommunications networks, multi-access edge computing, and security.