Network Virtualization and Technologies: SDN, NV, OpenFlow, VXLAN

Lecturer: PhD. Phan Xuan Thien

Nguyen Minh Thu
University of Information Technology – UIT
22521441
22521441@gm.uit.edu.vn

Duong Ba Can
University of Information Technology – UIT
22520143
22520143@gm.uit.edu.vn

Dinh Huynh Gia Bao
University of Information Technology – UIT
22520101
22520101@gm.uit.edu.vn

Tran Gia Bao
University of Information Technology – UIT
22520117
22520117@gm.uit.edu.vn

Abstract

The rapid development in the field of information technology has ushered in a new era for network virtualization—a technology that enables multiple logical networks to operate concurrently on a shared physical infrastructure. This study provides a comprehensive analysis of the modern network virtualization ecosystem, with a particular focus on foundational technologies including Software-Defined Networking (SDN), Network Functions Virtualization (NFV), the OpenFlow protocol, and VXLAN technology. By exploring the relationships among these components, the paper offers valuable insights into the transformative potential of network virtualization in optimizing performance, enhancing security, and increasing the flexibility of modern network infrastructures.

ACM Reference Format:
Nguyen Minh Thu, Duong Ba Can, Dinh Huynh Gia Bao, and Tran Gia Bao. 2025. Network Virtualization and Technologies: SDN, NV, OpenFlow, VXLAN: Lecturer: PhD. Phan Xuan Thien. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The wave of digital transformation is reshaping the very nature of higher education, compelling universities to evolve into intelligent ecosystems with exceptional adaptability. At the heart of this transformation lies the demand for a network infrastructure that is not only flexible and easy to manage but also seamlessly scalable to meet the increasingly complex needs of modern academic environments. While traditional network architectures have served effectively for decades, they are now revealing significant limitations in addressing emerging challenges such as multi-layered security, high automation requirements, and the need for performance optimization in the era of big data and cloud computing. In this context, two breakthrough technologies—Software-Defined Networking (SDN) and Network Virtualization—are emerging as pioneering solutions, offering centralized network management, flexible policy control, and superior cost-efficiency. This study explores the transformative potential of SDN and network virtualization in reshaping the IT infrastructure of universities. Integrating these technologies is not merely a technical advancement but a strategic enabler to enhance learning experiences, expand research capabilities, and optimize administrative operations—critical factors in the journey toward building truly smart universities in the digital age.

2 Overview

2.1 Virtualization Network

Network virtualization refers to the process of combining hardware (e.g., switches, routers) and software network resources into a single, software-based administrative entity. This abstraction allows for multiple, isolated virtual networks to coexist on the same physical infrastructure. [1]

2.2 SDN

Software-Defined Networking (SDN) is a networking approach that uses software controllers or APIs to manage and direct network traffic, as opposed to traditional networks that rely on dedicated hardware like routers and switches. SDN enables the creation and control of virtual networks or the management of traditional hardware through software. [2]

SDN structure includes:

2.2.1 Control plane SDN controllers handle communication with the apps to determine the destination of data packets. The controllers are the load balancers within SDN.[3]

2.2.2 Data plane Networking devices receive instructions from the controllers regarding how to route the packets. [3]

2.3 OpenFlow

OpenFlow is a standardized southbound interface used in Software-Defined Networking (SDN) that facilitates communication between a controller and forwarding elements. It allows devices to have multiple flow tables and enables header matching for packet forwarding based on specific actions. [4]

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.4 VXLAN

VXLAN is an extension of traditional VLAN technology. It allows the network to extend beyond local limits, in line with the trend of cloud development and modern network infrastructure.[5]

2.5 Challenges and Solutions

Table 1: Network Virtualization: Challenges and Solutions

Challenge Area	**Problem**	**Solution**
Scaling	Traditional VLAN technology limited to 4,096 virtual networks, insufficient for cloud environments	Overlay networks (VXLAN, NVGRE, STT, GENEVE) expand to 16 million virtual networks using tunneling techniques
Virtual Switching	Virtual machines require switching functions within hypervisors for communication	Software-based virtual switches (e.g., Open vSwitch) with flow tables and distributed control
Network Services	Traditional services (firewall, NAT, load balancing) must be integrated without performance loss	Network Function Virtualization (NFV) implements services in software with service chaining capabilities
Network State	Managing state for thousands of virtual networks while supporting VM mobility	Distributed controllers maintain synchronized network state; network hypervisors abstract physical infrastructure
Performance	Virtualization overhead impacts throughput and latency	Hardware offloading with specialized NICs; kernel bypass technologies (DPDK) for improved performance

Table 1 summarizes key challenges in virtualized networks such as scaling, switching, network services, network state, and performance have been effectively addressed by modern technologies such as overlay networks, SDN, NFV, and dedicated hardware. These solutions make networks more flexible, easier to manage, and more efficient, especially suitable for large-scale cloud and datacenter environments.

2.6 Architecture

Network virtualization fundamentally aims to provide abstraction of physical network resources, allowing multiple logical networks to coexist on a shared physical infrastructure. The architecture:[6]

2.6.1 *Basic Network Virtualization System*

- Creates isolated tenant networks with separate addressing schemes

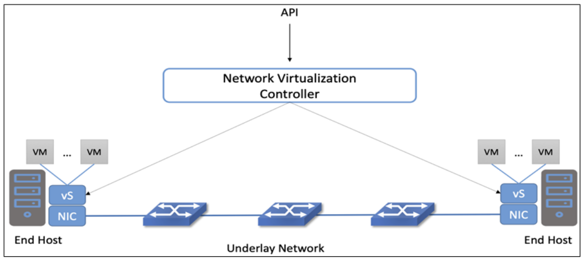


Figure 1: Simple Virtualization Network System

- Produces virtual topologies that can differ from the underlying physical network
- Network virtualization controller with northbound API, receives input describing the desired state of the virtual network

2.6.2 Virtual Overlay Encapsulation The virtual overlay model forms the core architectural approach:[6]

- Creates virtual networks as overlays on existing physical infrastructure
- Uses encapsulation (tunneling) to transport virtual network packets across physical networks
- Implements edge-based processing where physical network endpoints handle encapsulation/decapsulation
- Allows physical underlay network to remain unaware of the virtual networks it transports
- Provides scaling beyond traditional methods like VLANs (which are limited to 4096 identifiers)

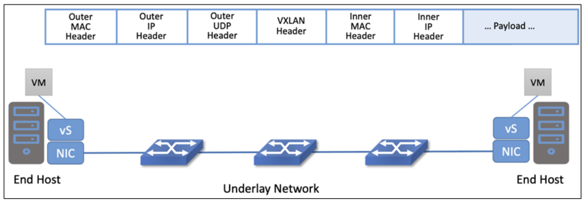


Figure 2: Encapsulation

2.6.3 Distributed Services Network virtualization architecture incorporates distributed network services:

- Logical routing implemented across distributed virtual switches
- Distributed firewalls enforcing security policies close to workloads
- Load balancing functionality distributed across edge devices
- Service chaining capabilities to connect network functions
- Centralized policy definition with distributed enforcement

2.6.4 Management, Control, and Data Plane The architecture follows a clear separation of concerns:[6]

Management Plane: Handles provisioning, configuration of policies, and resource allocation

Control Plane: Maintains mappings between virtual and physical components, distributes forwarding information

Data Plane: Performs actual packet forwarding, encapsulation/decapsulation, and policy enforcement

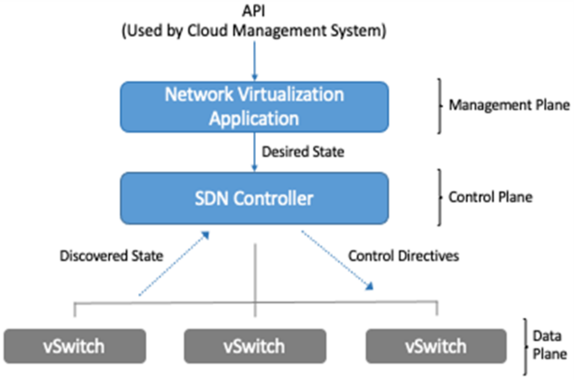


Figure 3: The three planes of a network virtualization system

This three-layer approach enables centralized management with distributed enforcement, improving both scalability and performance.

2.6.5 Encapsulation Protocol Key protocols enabling the architecture include:[6]

VXLAN: VLAN-like encapsulation for large-scale environments with 24-bit VNI (VXLAN Network Identifier)

GENEVE: Generic Network Virtualization Encapsulation providing extensibility through options

STT: Stateless Transport Tunneling designed for efficient software processing

NVO3: Network Virtualization Overlays standard encompassing various protocols

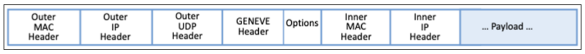


Figure 4: GENEVE header format

These protocols create tunnels between endpoints, carrying virtual network packets with identifiers to maintain isolation.

2.6.6 Virtual Switch Virtual switches serve as critical components:[6]

- Deployed at the network edge, typically within hypervisors
- Function as the virtual-physical boundary points in the architecture
- Perform encapsulation/decapsulation of traffic
- Apply security policies, QoS, and other network services
- Maintain mappings between virtual MAC addresses and physical locations
- Support standards like Open vSwitch Database (OVSDB) for configuration

2.6.7 OVN (Open Virtual Network)

- Provides logical switches and routers in a distributed virtual networking solution
- Implements the control plane for Open vSwitch
- Uses a centralized controller with distributed enforcement
- Supports L2/L3 abstractions with security groups and ACLs
- Integrates with cloud platforms like OpenStack
- Maintains logical network state in databases while pushing configurations to edge devices

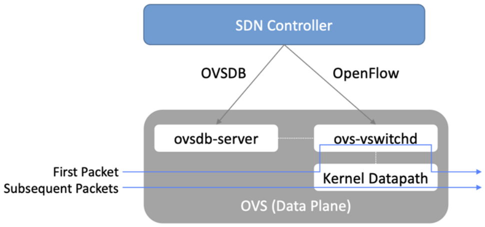


Figure 5: Function of Open switch blocks

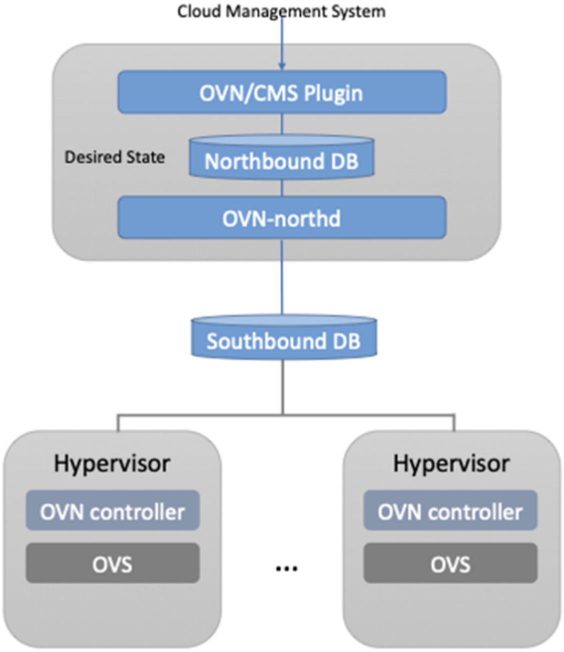


Figure 6: OVN High Level Architecture

This multi-layered architecture enables network virtualization that can scale to cloud environments while maintaining performance, security, and manageability.

2.7 Microsegmentation

2.7.1 Microsegmentation Network virtualization enables a change in how security is deployed in the data center:[6]

- Deploy security in a distributed, software-based manner
- Easily create multiple isolated networks

2.7.2 Compared to traditional segmentation

- Traditional segmentation: Group multiple machines into large "zones", using firewalls between zones
- Micro-segmentation: Create narrowly defined virtual networks, defining exactly which machines can communicate and how.

2.7.3 Advantages of micro-segmentation

- Fine-grained policy definition (e.g., a web-tier machine can only communicate with the application tier on specific ports)

- Easy migration of VMs without complex VLAN reconfiguration
- Improved security by minimizing the attack surface

3 System design

3.1 SDN project model

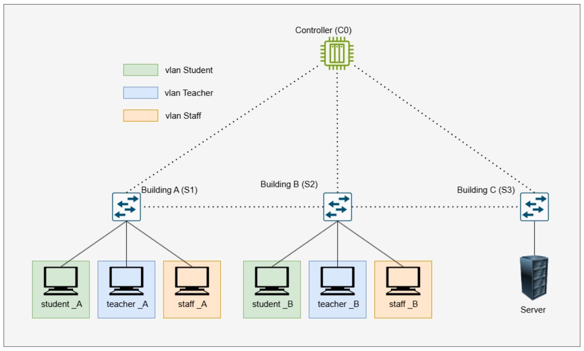


Figure 7: SDN project model

This figure shows the architecture of an SDN-based campus network divided into three buildings (A, B, and C), each managed by a local OpenFlow switch (S1, S2, S3). These switches are connected to a central SDN controller (C0), which controls network flows dynamically. Each building contains hosts assigned to different VLANs — students (green), teachers (blue), and staff (orange). Building C provides centralized server access. This setup enables efficient traffic management, segmentation by role, and centralized control for security and policy enforcement.

3.2 Scenario 1: Basic Network Virtualization Setup

- **Objective:** Create three separate virtualized networks for Student, Teacher, and Staff
- **Implementation:**
  - Use Mininet to simulate university network topology
  - Configure Ryu controller to create basic flow rules
  - Deploy VLAN to separate the three virtual network zones
- **Evaluation:** Test isolation and connectivity between network zones

3.3 Scenario 2: Simulate Response based on User Role

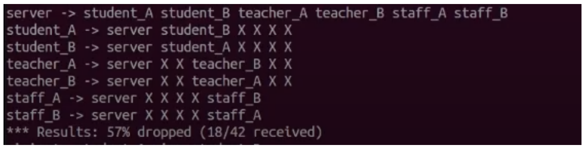
- **Objective:** Simulate the dynamic behavior of the server when users belong to different roles (Student, Teacher, Staff)
- **Implementation:**
  - Use Python to build a simple Flask server
  - The user sends a request (wget/curl) to the server
  - The server analyzes the role based on IP or URL and responds with appropriate messages
- **Evaluation:** Illustration of the application of user discrimination logic

3.4 Scenario 3: Bandwidth Management by User Group

- **Objective:** Distribute bandwidth by priority (Staff > Teacher > Student)
- **Implementation:**
  - Configure QoS on SDN controller
  - Bandwidth limitation by priority: Staff (15 Mbits/s) → Teacher (10 Mbits/s) → Student (5 Mbits/s)
- **Evaluation:** Measure the access speed of each hosts to the server
- **Note:** QoS is a mechanism that allows network administrators to control and optimize network traffic by assigning priorities to different types of data.

4 Experimental Results

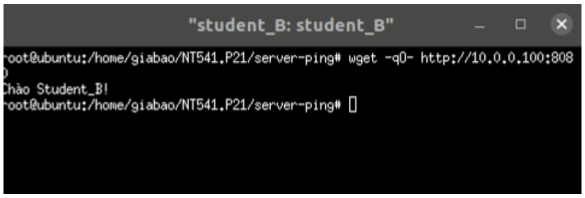
- **Scenario 1:** The research team successfully divided the network into VLANs and isolated different network segments. Additionally, the team verified connectivity between hosts within the same VLAN.



```
server -> student_A student_B teacher_A teacher_B staff_A staff_B
student_A -> server student_B X X X X
student_B -> server student_A X X X X
teacher_A -> server X X teacher_B X X
teacher_B -> server X X teacher_A X X
staff_A -> server X X X X staff_B
staff_B -> server X X X X staff_A
*** Results: 57% dropped (18/42 received)
```

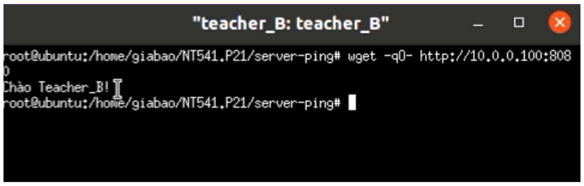
Figure 8: "pingall" command for checking

- **Scenario 2:** The research team successfully configured the server to respond with welcome messages based on the VLAN of the accessing host.



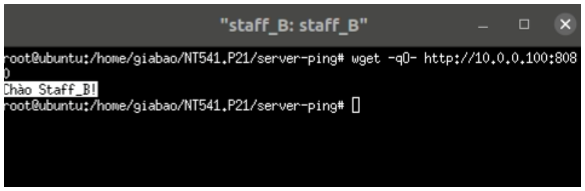
```
root@ubuntu:/home/giabao/NT541.P21/server-ping# wget -qO- http://10.0.0.100:808
0
Chào Student_B!
root@ubuntu:/home/giabao/NT541.P21/server-ping# 
```

Figure 9: Connect to server from VLAN Student



```
root@ubuntu:/home/giabao/NT541.P21/server-ping# wget -qO- http://10.0.0.100:808
0
Chào Teacher_B!
root@ubuntu:/home/giabao/NT541.P21/server-ping# 
```

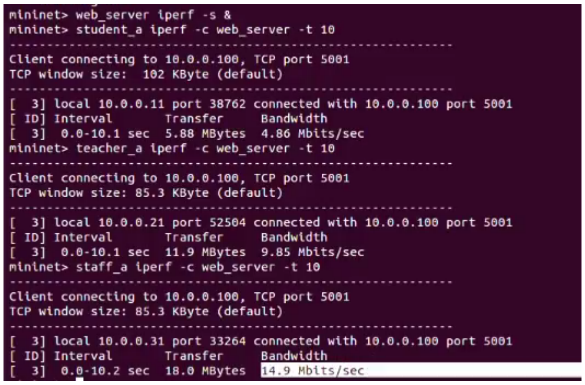
Figure 10: Connect to server from VLAN Teacher



```
root@ubuntu:/home/giabao/NT541.P21/server-ping# wget -qO- http://10.0.0.100:808
0
Chào Staff_B!
root@ubuntu:/home/giabao/NT541.P21/server-ping# 
```

Figure 11: Connect to server from VLAN Staff

- **Scenario 3:** The research team completed the allocation of server access bandwidth for each host based on the VLAN classification of that host.



```
mininet> web_server iperf -s &
mininet> student_a iperf -c web_server -t 10
------------------------------------------------------------
Client connecting to 10.0.0.100, TCP port 5001
TCP window size:  102 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.11 port 38762 connected with 10.0.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec  5.88 MBytes  4.86 Mbits/sec
mininet> teacher_a iperf -c web_server -t 10
------------------------------------------------------------
Client connecting to 10.0.0.100, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.21 port 52504 connected with 10.0.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec  11.9 MBytes  9.85 Mbits/sec
mininet> staff_a iperf -c web_server -t 10
------------------------------------------------------------
Client connecting to 10.0.0.100, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.31 port 33264 connected with 10.0.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.2 sec  18.0 MBytes  14.9 Mbits/sec
```

Figure 12: Check throughput of each host

5 Final comparison

Table 2 summarizes:

- Virtualization Network in SDN represents a revolutionary approach that separates control from data transmission layers. It manages the entire network through centralized software controllers, enabling programmable network behavior, flexible policy management, and easier large-scale expansion.
- Traditional Virtualization Network creates virtual network layers over existing physical infrastructure, improving flexibility without changing how the underlying network fundamentally works. It maintains dependence on hardware devices while enabling VM mobility.
- Key Difference: Traditional virtualization works on top of existing networks, while SDN transforms how networks function at their core.
- This comparison reveals that neither approach is universally superior - the choice depends on organizational needs, existing infrastructure, skills, and long-term network strategy.

Table 2: Compare Virtualization Network and SDN

Criteria	**Virtualization Network in SDN**	**Traditional Virtualization Network**
Concept	Network virtualization through a software-based, programmable architecture that decouples network control from underlying hardware	Network virtualization that creates logical network layers on top of existing physical network infrastructure
Control plane	Centralized software controller with full programmatic control and network-wide visibility	Distributed control across individual network devices with limited centralized management
Network structure	Logically centralized control with abstracted network resources that can be dynamically configured.	Hierarchical network structure with fixed configurations tied to physical hardware.
Switching equipment	Software-defined switching with dynamic flow tables and programmable packet forwarding	Hardware-based switching with fixed routing and predefined network paths
Flexibility and mobility	Extremely high flexibility with real-time network reconfiguration and dynamic resource allocation	Moderate flexibility, limited by hardware constraints and manual configuration processes
Control interface	Open, programmable APIs allowing direct software-based network management and automation	Vendor-specific interfaces with limited programmability and more manual intervention
Extensibility	Highly extensible through software updates, easy to add new services and network functions	Limited extensibility, typically requires hardware upgrades or replacements
Main difference	Transforms network functionality through software-defined approach, enabling programmable and adaptive networking	Adds virtual layers on existing network infrastructure without fundamentally changing network operations

6 Limitation and Future work

Limitation in the current approach and implementation of network virtualization and SDN technologies:

- **Limited real-world deployment:** While SDN and network virtualization are theoretically sound and have been successfully simulated, their deployment in real academic or enterprise environments is still limited due to infrastructure

costs, legacy compatibility issues, and the need for skilled personnel.
- **Performance constraints:** Virtualization introduces performance overhead, especially in high-throughput or latency-sensitive applications. Although hardware offloading and protocols like DPDK help, more robust solutions are required for mission-critical networks.
- **Security challenges:** Microsegmentation and virtual firewalls improve security, but managing distributed policies and detecting threats in dynamic virtual environments remains a challenge. Integration of AI-based threat detection could enhance resilience.
- **Controller bottlenecks:** The centralized nature of SDN controllers may lead to single points of failure or performance bottlenecks. Future architectures should explore distributed or hierarchical controller models.
- **Lack of interoperability standards:** Diverse implementations of OpenFlow, VXLAN, and other protocols often lack standardization, limiting interoperability between vendors and platforms.

Future work can focus on several key areas:

- Expanding experimental validation with real hardware in campus or enterprise testbeds to assess scalability, QoS, and failure recovery.
- Integrating AI and ML techniques for dynamic policy enforcement, anomaly detection, and intelligent traffic routing.
- Enhancing cross-layer security frameworks tailored for virtualized networks and SDN controllers.
- Exploring hybrid models combining SDN with intent-based networking and network slicing to support multi-tenant environments and 5G infrastructures.
- Developing standardized APIs and interfaces to improve integration and interoperability across platforms and vendors.

7 Conclusion

In this paper we presented SDN and Virtualization Network that changing the way modern infrastructure networks are built. Delivering flexible, scalable, easy-to-manage and highly secure systems. Ready for chemical trends such as Cloud, IoT, Smart Campus, ... Simulation scenarios illustrate virtual network segmentation, user role analysis capabilities. Creating the premise for access control, resource segmentation and intelligent services [1]

References

[1] H. L. Stanforth, "Virtualization: A Review and Future Directions - Executive Overview," ResearchGate, Accessed: May 11, 2025

[2] VMware, "Software-Defined Networking," VMware, Accessed: May 11, 2025.

[3] IBM, "Software-Defined Networking," IBM, Accessed: May 11, 2025.

[4] ScienceDirect, "OpenFlow," ScienceDirect, Accessed: May 11, 2025.

[5] Cisco, "Introduction to VXLAN," Cisco Learning Network, Accessed: May 11, 2025.

[6] L. Peterson and B. Davie, Network Virtualization, SystemsApproach.org. Accessed: May 11, 2025. [Online]. Available: https://sdn.systemsapproach.org/netvirt.html

[7] M. T. Arashloo, CS 856: Programmable Networks, Lecture 1: SDN and OpenFlow, lecture, Winter 2023.

[8] K. Smiler, OpenFlow Cookbook, Packt Publishing, Apr. 2015. [Online]. Available: https://www.everand.com/book/272071499/OpenFlow-Cookbook. Accessed: May 11, 2025.

[9] M. Casado, T. Koponen, D. Moon, and S. Shenker, "Rethinking Packet Forwarding Hardware," IEEE Micro, vol. 37, no. 2, pp. 12–21, Mar.–Apr. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8066289. Accessed: May 11, 2025.