

Enhancing Security in Software-Defined Networking (SDN) based IP Multicast Systems: Challenges and Opportunities

UNIVERSITY OF TURKU
Department of Computing
Master's of Science in Technology Thesis
Cybersecurity
May 2024
Preety Prasad

Supervisors:
Tahir Mohammad
Petri Sainio

UNIVERSITY OF TURKU
Department of Computing

PREETY PRASAD: Enhancing Security in Software-Defined Networking (SDN) based
IP Multicast Systems: Challenges and Opportunities

Master's of Science in Technology Thesis, 85 p., 3 app. p.
Cybersecurity
May 2024

This thesis explores how software-defined networking (SDN) can enhance security in IP multicast, leading to efficient and secure content delivery within the broadcasting industry. Traditional multicast offers efficient data transmission but faces limitations in security and scalability. SDN-based IP multicast addresses these challenges through centralized control and programmability. However, new security threats emerge with SDN.

This thesis proposes a framework for real-time event monitoring specifically tailored for securing IP multicast traffic within SDN environments. This framework leverages real-time analysis and pre-defined event monitoring policies to detect and mitigate security threats such as DoS attacks and unauthorized access. Additionally, the framework explores a mechanism for integrating with the SDN controller, enabling it to leverage the benefits of centralized control for automated responses like traffic rerouting or flow rule updates.

The evaluation of the proposed framework demonstrates its effectiveness in improving security posture and reducing the risk of successful attacks on multicast communication. This approach, combined with robust security practices, ultimately enhances viewer experiences by ensuring the integrity, confidentiality, and availability of network resources and data transmission in modern broadcast environments.

Keywords: Multicast, data transmission, efficiency, bandwidth optimization, routing protocols, Software-Defined Networking (SDN), scalability, cybersecurity, threat attacks, routing protocol.

Acknowledgements

I am profoundly grateful to extend my sincere appreciation to my supervisor **Tahir Mohammad** and co-supervisor **Petri Sainio** for their exceptional mentorship and profound academic insights, particularly within the realm of Cybersecurity during my master's program in ICT. Their passion for the advanced technologies in ICT and commitment to excellence served as a constant source of inspiration, propelling me to explore and delve deeper into this advanced domain. Their encouragement and guidance have significantly contributed to my academic and intellectual growth, laying a solid foundation for the endeavours ahead.

Additionally, I wish to express my heartfelt gratitude to the esteemed supervisors and dedicated staff members at **Yleisradio Oy**. Their unwavering support, patience, and invaluable guidance have been instrumental throughout the journey of crafting this master's thesis. Their expertise not only enriched the technical and organizational facets of my work but also fostered an environment where creativity in technology and innovation thrived. Their mentorship has been a cornerstone in shaping both the content and the approach of this research, and for that, I am deeply indebted.

In reflecting upon this academic journey, I am reminded of the invaluable contributions and unwavering support of all those who have played a role, no matter how small. To each and every individual who has lent their expertise, offered encouragement, or shared in the joy of discovery, I extend my heartfelt gratitude. This thesis stands as a testament to the collective effort and dedication of a community committed to the pursuit of knowledge and innovation.

Contents

1	Introduction	1
1.1	Problem Statement	4
1.2	Motivation	4
1.3	Research Objectives	5
1.4	Research Questions	6
1.5	Thesis Organization	7
2	Literature Review	8
2.1	Types of streaming in IP multicast	8
2.2	Applications	9
2.3	Traditional IP Multicast	10
2.3.1	Multicast Routing	10
2.3.2	Routing Protocols	11
2.3.3	Flooding schemes for traditional multicast system	12
2.3.4	Forwarding traffic with Rendezvous Point (RP)	13
2.3.5	PIM-SM Source Specific Multicast	15
2.3.6	Bidirectional PIM	16
2.3.7	Link state multicast routing	16
2.3.8	Core-Based Trees and PIM-SM	16
2.3.9	Other relevant issues	17

2.3.10	Delivering multicast service using IGMP	18
2.3.11	IGMP Snooping	19
2.3.12	Layer 3 vs Layer 2 multicast	19
2.3.13	STP Protocol	20
2.4	Time management protocols	20
2.5	Security issues in traditional IP multicast	21
2.5.1	Threats against Routers	21
2.5.2	Network Attacks on IGMPv3	22
2.5.3	Network Attacks on PIM	24
2.5.4	Other drawbacks	25
2.6	Overview of SDN Based IP Multicast	25
2.6.1	Software-Defined Networking	26
2.7	SDN Architecture	29
2.7.1	Overview of SDN security features	31
2.7.2	Benefits of Network security	33
2.7.3	SDN: Revolutionizing Information Security	35
2.8	Seamless Switching for Media Streams: SMPTE 2022-7	36
2.8.1	Red and Blue networks	37
2.8.2	Functioning of redundant IP packets	38
2.9	Multicast Support in EVPN-VXLAN	40
2.10	Conclusion	40
3	Cyberattacks in SDN	43
3.1	Attacks on Infrastructure Plane	44
3.2	Attacks on Control Plane	45
3.3	Attacks on Application Plane	46
3.4	Attacks on the SDN interfaces	48
3.5	Miscellaneous Threats	49

3.6	Topology Discovery Threats	49
3.7	Conclusion	52
4	Limitations in current SDN Solutions	53
4.1	Security Limitations	53
4.2	Scalability Challenges	54
4.3	Monitoring and Analytics	54
4.4	Automation and Orchestration Focus	54
4.5	Conclusion	55
5	Recommendations for Mitigation Strategies	56
5.1	Network segmentation	56
5.2	Detection and response mechanism using devices	57
5.3	Monitoring network connectivity by updating flow graph	57
5.4	Network protection against LLDP-based attacks	58
6	Proposed Solution: Real-time event monitoring policies framework	60
6.1	Related work	61
6.2	Framework: SDN-Multicast Security Monitor Policies	65
6.2.1	Advanced Techniques	66
6.2.2	Threat Modeling Integration and Continuous Improvement	67
6.2.3	ML, Real-time Monitoring, and Threat Intelligence	68
6.2.4	SMSMP Application Model integrated with SDN	69
6.2.5	Key Advantages of SMSMP	73
6.2.6	Implementation of SMSMP	75
6.2.7	Tools for implementing	78
6.2.8	Success Factors Metrics	80
7	Conclusion	82

References	85
Appendices	
A ARISTA Multicast Orchestration	A-1
B MCS API workflow	B-1

List of Figures

2.1	Unicast and Multicast transmission	9
2.2	PIM Sparse Mode	14
2.3	Traditional Networking vs Software Defined Networking	27
2.4	SDN with Openflow controller Architecture	30
2.5	SMPTE 2022-7 Standard [9]	39
6.1	Threat modelling framework for SDN incorporating SMSMP	69
6.2	SMSMP integrated with SDN	70
A.1	ARISTA MCS across deployments [44]	A-1

List of Tables

2.1	SDN for Enhanced Security: Exploring Control and Visibility Features	32
3.1	SDN Topology discovery threat attacks	50
6.1	Description of Modules and Tools	79
6.2	Success Factor Matrix for SMSMP Security Framework	80

Acronyms

ACL Access Control List

API Application Programming Interface

ASM Any Source Multicast

CBT Core Based Trees

CPU Central Processing Unit

DDos Distributed Denial of Service

DM Dense Mode

DNS Domain Name System

DPI Deep Packet Inspection

DoS Denial of Service

DVMRP Distance Vector Multicast Routing Protocol

EOS Extensible Operating System

EVPN Ethernet VPN

FM Frequency Modulation

HLS HTTP Live Streaming

HTTP Hypertext Transfer Protocol

HTS Host Tracking System

IGMP Internet Group Management Protocol

IGMPv3 Internet Group Management Protocol version 3

IDS Intrusion Detection System

IPS Intrusion Prevention System

IP Internet Protocol

IPsec Internet Protocol Security

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

JSON JavaScript Object Notation

LAN Local Area Network

L2 Layer 2

L3 Layer 3

LLDP Link Layer Discovery Protocol

LSTM Long Short-Term Memory

LFA Link Fabrication Attack or Link Flooding Attacks

MAC Media Access Control

MBGP Multiprotocol Border Gateway Protocol

MDDT Multicast Data Distribution Tree

MD5 Message Digest Algorithm

MITM Man In The Middle

MCS Media Control Services

MPLS Multiprotocol Label Switching

MOSPF Multicast Open Shortest Path First

MRIB Multicast Routing Information Base

MLD Multicast Listener Discovery

MTBF Mean Time Between Failures

MTTR Mean Time to Resolution (MTTR)

NTP Network Time Protocol

OB Outside Broadcast

ODL OpenDaylight Project

OF OpenFlow

OFDP OpenFlow Discovery Protocol

OpenSDWN Open Software-Defined Wireless Networking

PBR Policy-Based Routing

PIM Protocol Independent Multicast

PT Precision Time Protocol

QoS Quality of Service

RAM Random Access Memory

RFC Request for Comments

RBAC Role-Based Access Control

REST Representational State Transfer

RPF Reverse Path Forwarding

RP Rendezvous Point

RPT Rendezvous Point Trees

RTP Real-time Transport Protocol

RTSP Real-Time Streaming Protocol

SHA Secure Hash Algorithm

SDN Software Defined Networking

SDP Session Description Protocol

SIP Session Initiation Protocol

SRT Secure Reliable Transport

STIX Structured Threat Expression Information

SNMP Simple Network Management Protocol

SMSMP SDN-Multicast Security Monitor Policies

SSM Source Specific Multicast

SVM Support Vector Machine

SIDR Security Incident Detection Rate

SIEM Security Information and Event Management

SSL Secure Sockets Layer

SMPTE-ST Society of Motion Picture and Television Engineers Standard

SNMP Simple Network Management Protocol

SSDP Simple Service Discovery Protocol

TAXII Trusted Automated Exchange of Indicator Information

TCP Transmission Control Protocol

TLS Transport Layer Security

TTD Time to Detection

TV Transport Stream Video

UDP User Datagram Protocol

VLAN Virtual Local Area Network

VXLAN Virtual Extensible LAN

WAN Wide Area Network

1 Introduction

Multicast technology offers a sophisticated approach to data transmission, particularly beneficial in scenarios where content needs to be efficiently distributed to multiple recipients simultaneously. Unlike unicast and broadcast methods, multicast optimizes network bandwidth usage by transmitting data from a single source to a specific group of recipients, called a multicast group. This targeted approach minimizes network congestion and enhances overall efficiency, making it ideal for applications like streaming media and online gaming.

Traditional networking relies on fixed-function and dedicated hardware, including switches, routers, and application delivery controls, to manage network traffic effectively. These devices perform specific functions and work together seamlessly to support network operations. However, scalability is often a challenge as switching hardware and software is typically proprietary, lacking Application Programming Interface (API) for easy provisioning. While traditional networks synergize well with proprietary provisioning software, they face limitations due to the inability to modify such software and the hardware-centric nature of networking, primarily relying on application-specific integrated circuits (ASIC) and other dedicated hardware for functionality implementation.

The most commonly used multicast protocols for live video streaming include HTTP Live Streaming (HLS), Secure Reliable Transport (SRT), Real-Time Streaming Protocol (RTSP), TCP, UDP, and SDP.

While not cutting-edge by today's standards, Protocol Independent Multicast (PIM) Dense Mode (DM) is a veteran of the IP multicast scene. Developed in the early 1990s, PIM-DM has proven itself as a reliable and scalable solution for networks with many receivers. Its compatibility with a wide range of network devices makes it a familiar and trusted choice for network administrators.

PIM-DM works by initially flooding multicast traffic throughout the network. Devices that don't have interested receivers then send "prune" messages upstream, effectively trimming unnecessary branches from the multicast distribution tree. This flood-and-prune approach ensures efficient multicast delivery but can be resource-intensive in large or dynamic networks.

In essence, PIM-DM offers a simple and reliable multicast solution, but its reliance on manual configuration and potential for bandwidth consumption in certain scenarios might necessitate exploring alternative approaches for some networks.

Multicast technology, whether operating at the network layer with router support or as an application-level overlay network leveraging regular hosts for routing and switching, offers efficient data dissemination to multiple recipients. The most important routing protocols for multicast, including Dense Mode (DM) and Sparse Mode (SM), facilitate this process, by flooding the network using DM with multicast traffic and SM operating on a pull mechanism. Implementing multicast involves consideration of routing algorithms such as source-specific trees and shared trees to optimize data transmission paths. Additional protocols like bidirectional PIM and source-specific multicast (SSM) further enhance efficiency in many-to-many applications. However, multicast technology faces challenges including security vulnerabilities, scalability limitations, and dependence on specialized protocols for reliable transmissions, with traditional IP multicast protocols such as IGMPv3 susceptible to various network attacks.

Software-Defined Networking (SDN) has emerged as a technology attracting signifi-

cant attention in recent years. The dynamic landscape of modern IT infrastructures, driven by SDN and its projected market value of \$25.3 billion by 2026 [1], offers a compelling platform for academic inquiry and practical application. SDN's separation of control from hardware empowers researchers and practitioners to design innovative network functions and protocols with greater flexibility and ease. This new technology presents exciting research opportunities for network security. SDN has the potential to greatly impact security research in various aspects.

SDN-based multicast is an innovative approach that offers faster convergence and improved scalability compared to traditional methods for IP multicast. Its centralized control simplifies management, making it a tempting option for administrators seeking a more streamlined approach.

Exploring how SDN's features like centralized control improve network and information security, this thesis offers valuable insights for future research in this crucial area.

SDN entails separating the control and packet forwarding planes in a network, facilitating connectivity with applications via APIs. This relationship enhances application performance and security, fostering a scalable and dynamic network architecture. Widely adopted by enterprises worldwide for application deployment, SDN streamlines deployment processes and reduces operational costs. With centralized provisioning and management capabilities, SDN optimizes network resources, offering programmatic control through open APIs. By separating network configuration and traffic management from the underlying hardware, SDN facilitates the use of open protocols such as OpenFlow. This allows access to network devices that usually depend on proprietary firmware. This method utilizes globally aware software control, leading to improved network flexibility and management.

The emergence of SDN offers a promising solution to address the shortcomings of traditional multicast approaches. SDN separates the control and data planes, providing

centralized control and programmability, which enhances network management flexibility and scalability. By utilizing SDN-enabled IP multicast systems, broadcasters can achieve enhanced security, multicast addressing efficiency, and dynamic flow orchestration.

1.1 Problem Statement

Traditional IP multicast offers significant advantages for broadcasting services but is hampered by limitations in security, scalability—especially with large groups—and reliable transmission mechanisms. Protocols like IGMPv3 and PIM are vulnerable to attacks such as source spoofing and denial-of-service (DoS), which necessitate complex security measures like cryptography and access controls. Furthermore, these protocols often struggle to handle network failures and adapt to evolving security threats effectively. SDN, powered by OpenFlow, provides a promising solution to these issues through centralized control and programmability. This research aims to explore the potential of SDN to enhance the security and scalability of multicast communication, focusing on addressing source authentication issues and mitigating eavesdropping attacks.

1.2 Motivation

Traditional IP multicast architectures face additional challenges due to their reliance on hardware-based routing decisions, leading to limited scalability and flexibility. These limitations expose networks to unauthorized access and manipulation, partly because of their lack of control and visibility. While SDN offers a software-defined approach with centralized control, this also introduces new security vulnerabilities through potential attack surfaces on the control plane. The rigidity of traditional multicast and the new risks in SDN-based solutions necessitate the development of

frameworks that can address these security gaps. To meet this need, the proposed SDN-Multicast Security Monitor Policies (SMSMP) framework integrates real-time event monitoring with centralized SDN control to enhance security incident detection and response mechanisms. By incorporating machine learning and encryption techniques, SMSMP aims to create a more secure and scalable multicast architecture. This research will evaluate the impact of this framework on security and network performance, with the goal of offering a more robust and adaptable solution for modern broadcast environments.

1.3 Research Objectives

1. **Evaluate Security Risks in SDN Transition:** To assess the security risks associated with transitioning from traditional IP-based broadcast networks to SDN-enabled architectures, focusing on potential vulnerabilities and threats in SDN controllers.
2. **Analyze Security Solutions for SDN-based SSM:** To examine the effectiveness of different security solutions for Source-Specific Multicast (SSM) with Media Control Services (MCS) in a spine-leaf network architecture.
3. **Recommend Security Solutions and Mitigation Strategies:** To identify effective solutions and mitigation strategies for the security issues found during the analysis.
4. **Propose a Real-Time Monitoring Framework:** To develop a framework named SDN-Multicast Security Monitor Policies (SMSMP) for real-time event monitoring and threat detection, using machine learning algorithms trained on historical traffic data and threat intelligence feeds.

1.4 Research Questions

1. How do the limitations of traditional IP multicast routing protocols compare to the capabilities of a centralized SDN architecture for managing multicast traffic in broadcast networks?
2. Given the identified security vulnerabilities in SDN-based network architecture, what mitigation strategies can be implemented, and what opportunities might arise from exploiting these vulnerabilities?
3. What are the benefits and challenges of integrating a real-time event monitoring system within the SDN controller?

1.5 Thesis Organization

- Chapter 1:** Introduces, motivates, and outlines the statement of objectives for the thesis.
- Chapter 2:** Offers an overview of IP multicasting and the main concepts behind Software-Defined Networking (SDN) and its security concerns.
- Chapter 3:** Analyzes the vulnerability and security threats in SDN-based IP multicasting.
- Chapter 4:** Describes the limitations and challenges of commercially available OpenFlow-based SDN solutions.
- Chapter 5:** Provides recommendations for the solution and mitigation strategies based on the analysis of chapter 3 and 4.
- Chapter 6:** Proposes a framework model for real-time event monitoring policies to detect and mitigate security threats.
- Chapter 7:** Provides the conclusion and points out possible areas of future work.

2 Literature Review

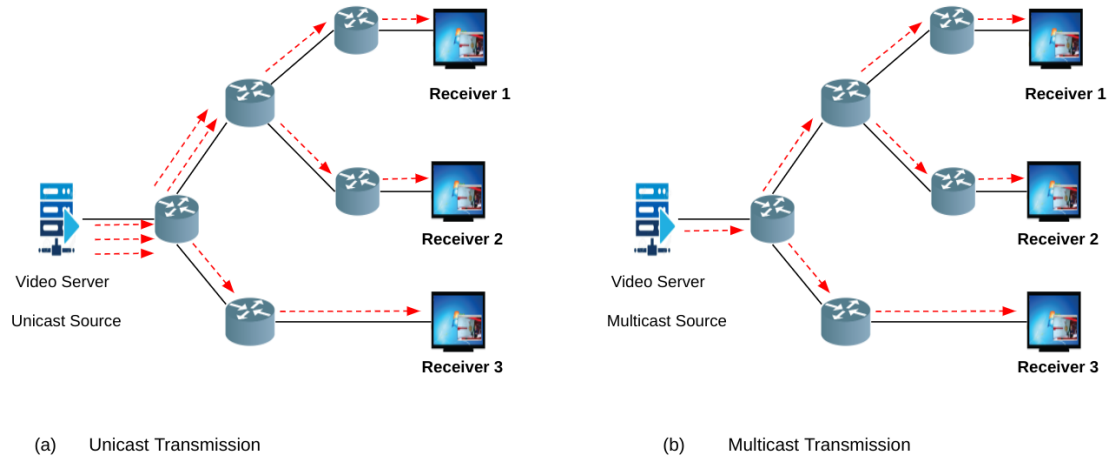
This section provides detailed insights into IPv4 Multicast implemented using dynamic routing protocols, the SDN-enabled IPv4 Multicast system, and the various threats encountered by its layered architecture.

2.1 Types of streaming in IP multicast

Unicast streaming operates on a one-to-one transmission model, with data transmitted from a single source to a solitary destination, ensuring each recipient receives its own distinct stream of data. This method involves sending data individually to each recipient, necessitating a separate stream for every viewer, potentially leading to higher bandwidth utilization compared to multicast streaming, as shown in Figure 2.1(a). For instance, video-on-demand platforms such as Netflix or YouTube employ unicast streaming, where each user's device receives a personalized data stream from the platform's servers when watching a video.

Multicast streaming efficiently delivers data from one source to multiple recipients simultaneously, minimizing bandwidth usage. This method works by transmitting data once from the source to a group of interested receivers, with network routers replicating and distributing the content as necessary, as shown in Figure 2.1(b). For instance, live video feeds like sports events or online conferences can be seamlessly distributed to numerous viewers over the internet. In contrast, unicast streaming involves a separate data stream for each recipient, potentially resulting in higher

Figure 2.1: Unicast and Multicast transmission



bandwidth consumption and less efficient distribution.

Management streaming involves transmitting control and monitoring data for overseeing network devices and services. It facilitates the exchange of configuration updates, performance metrics, error logs, and status information between devices and management systems. SNMP is an example of a protocol that enables remote monitoring and management of routers, switches, and servers by exchanging information with a centralized system, enhancing network administration efficiency.

Multicast can operate natively at the network layer if supported by routers or as an application-level overlay network, where regular hosts handle routing and switching.

2.2 Applications

Multicast serves as an ideal solution for applications that benefit from grouping hosts logically, particularly when large data volumes need simultaneous transmission to multiple recipients. This approach offers notable advantages over unicast, including bandwidth conservation and a reduced workload. Common applications benefiting from multicast include multimedia scenarios such as TV channel live streaming.

Multicast also presents a viable alternative to broadcast in scenarios where the sender is unaware of the receiver's identity or quantity, as seen in discovery protocols like SSDP and data center monitoring tools. Unlike broadcast, multicast offers controlled distribution, ensuring that hosts unrelated to a specific service or protocol do not receive irrelevant traffic, as multicast traffic can be filtered by routers, switches, and NICs, unlike broadcast messages, which are flooded indiscriminately.

2.3 Traditional IP Multicast

Effective implementation of multicasting involves four key steps:

- 1) The group owner creates a multicast host group with address and communicates this address to potential users.
- 2) Users or hosts interact with their local access router to join or leave the multicast group at the network level.
- 3) Routers establish links between the access router and the Multicast Data Distribution Tree (MDDT) through participation in a multicast routing protocol, with the construction specifics dependent on the chosen protocol.
- 4) Application-level protocols manage multicast data distribution during sessions, coordinating and allocating network resources like bandwidth and router configuration to optimize traffic flow [2].

2.3.1 Multicast Routing

Edge routers leverage the Internet Group Management Protocol (IGMP) to identify devices interested in specific multicast data. They then coordinate with other routers using protocols like RPF (Reverse Path Forwarding) to create efficient paths for sending that data. However, if some routers within the network can't directly manage multicast traffic, the data is encapsulated within regular data packets using

a unique method, like tunneling, to ensure it reaches its intended recipients. This encapsulation allows routers that don't understand multicast to forward the data based on the routing information of the outer packet.

Multicast routing protocols construct two main types of distribution trees: source-based trees and shared trees (explained in Section 2.3.7).

2.3.2 Routing Protocols

Multicast IP routing protocols facilitate the dissemination of data, such as audio or video streaming broadcasts, to numerous recipients simultaneously. Leveraging multicast capabilities, a sender can transmit a solitary data copy to a designated multicast address, subsequently reaching an entire group of recipients. These routing protocols operate at both Layer 2 and 3. Layer 3 includes IGMP, MLD, PIM, MSDP, and MBGP, which work for both IPv4 and IPv6. Layer 2 protocols like IGMP snooping, PIM snooping, and multicast VLANs complement Layer 3 protocols for efficient multicast forwarding within a network segment.

The Protocol-Independent Multicast (PIM) routing protocol comes in two modes: Dense Mode (PIM-DM) and Sparse Mode (PIM-SM). PIM-SM [RFC4601] optimizes traffic by building targeted trees for specific multicast groups, while PIM-DM assumes receivers are likely present on most subnets and forwards packets more broadly [3].

- *PIM-DM*: The protocols include distance vector multicast routing protocol (DVMRP) and PIM-DM employ shortest path trees (SPTs) to distribute multicast traffic based on the push principle. Figure 2.2 (b). This approach treats all subnets as if they contain devices interested in multicast traffic. Consequently, it replicates the data packets across the entire network, potentially leading to bandwidth congestion. This can be likened to FM radio broadcasting, where the signal is accessible in all areas and interested listeners tune into

the appropriate frequency [4].

- *PIM-SM*: Sparse mode, as the name suggests, ensures that the volume of traffic is relatively low compared to dense mode protocols, which flood the network with IP packets to transmit multicast traffic. Sparse mode protocols operate on a pull mechanism, where traffic is drawn from the receivers upon request, instead of being available at all nodes. This request from the receiver is transmitted through an explicit join mechanism, preventing unnecessary multicast traffic from wandering the network in search of recipients. While most dense mode protocols utilize SPTs to route traffic across the multicast network, in contrast, sparse mode protocols primarily rely on shared trees for multicast traffic distribution. PIM-SM, for example, can also leverage SPTs under specific conditions [4].

DM protocols are best suited for Local Area Network (LAN) environments where receivers are densely grouped and the bandwidth can handle flooding. On the other hand, SM protocols are typically more suitable for WAN environments. One of the earliest multicast routing protocols is the DVMRP [RFC1075]. It builds source-based trees using RPF and pruning mechanisms to efficiently deliver multicast traffic.

2.3.3 Flooding schemes for traditional multicast system

In PIM Dense mode, flooding schemes play a crucial role in distributing multicast traffic within a network. Upon receiving a multicast packet, a router forwards it on all its outgoing interfaces except the one it arrived on. This prevents the creation of infinite loops where the packet keeps circling back on itself. This process ensures that multicast traffic is flooded across the entire network, reaching all potential receivers. However, flooding can lead to inefficiencies and potential issues such as network congestion and unnecessary bandwidth usage, especially when multicast packets are forwarded to areas where there are no interested receivers. In multicast

communication, a source sends data to a group of interested receivers identified by a single multicast address. Multicast routers need to determine the direction of the source (upstream) and where receivers reside (downstream). When multiple paths lead downstream, the router replicates the packet and forwards it on appropriate downstream interfaces. This may not involve all downstream paths, depending on the specific multicast routing protocol used. This notion of directing multicast traffic away from the source instead of towards the receiver is referred to as RPF.

In multicast routing (like PIM Dense mode), Reverse Path Forwarding (RPF) prevents loops and ensures efficient delivery. RPF checks the incoming multicast packet's arrival interface against the best route for reaching the source (based on the unicast routing table). If the interfaces match, the packet is forwarded towards receivers (avoiding loops). If not, the packet is discarded to prevent forwarding potentially looped or invalid multicast traffic.

Overall, flooding schemes in PIM Dense mode enable multicast traffic to be distributed across the network, while RPF ensures loop-free packet forwarding by verifying the path of incoming multicast packets. These mechanisms work together to facilitate efficient and reliable multicast communication within a network.

2.3.4 Forwarding traffic with Rendezvous Point (RP)

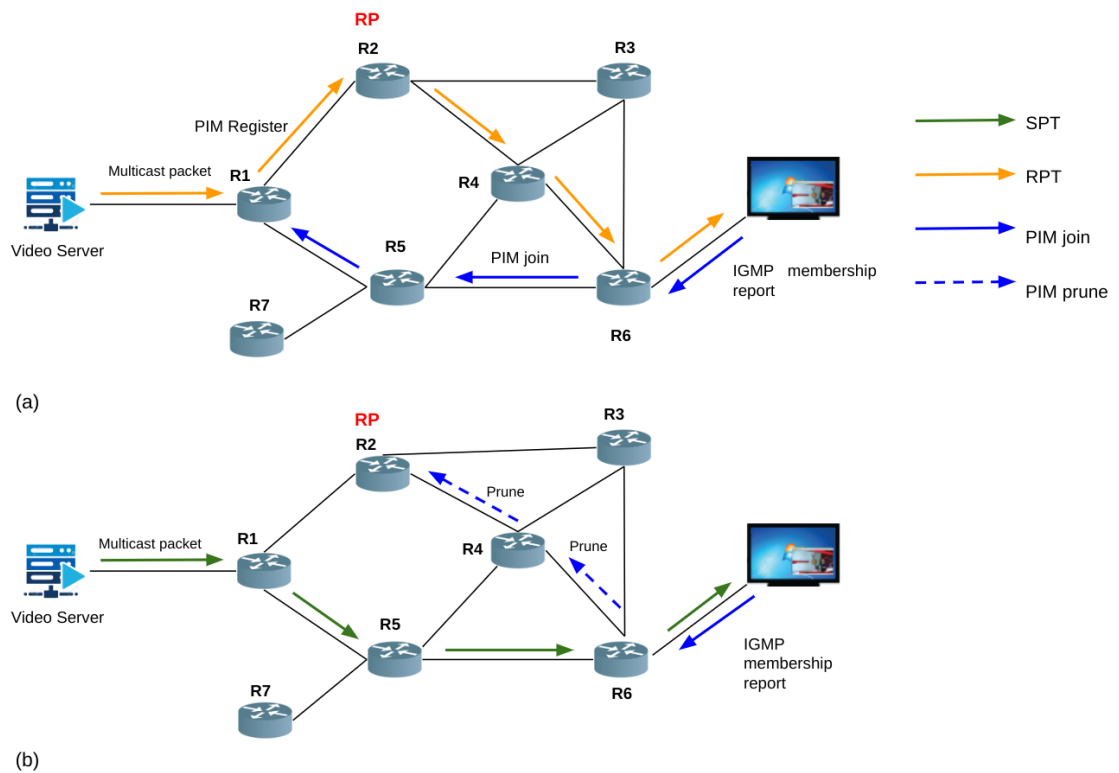
In PIM-SM, the approach to multicast routing differs from that of PIM-Dense mode. Instead of flooding multicast traffic throughout the network, PIM-SM establishes forwarding paths only when there are interested receivers for specific multicast groups. This conserves network bandwidth and reduces unnecessary traffic.

In PIM-SM, a rendezvous point (RP) acts as a central coordinator for multicast communication. Sources wishing to send multicast traffic to a specific group first register with the RP. Likewise, receivers interested in joining that group send a join message to the RP. This process allows PIM-SM to build a shared distribution

tree, also known as the Rendezvous Point Tree (RPT). This tree efficiently delivers multicast traffic from the RP to all receivers in the group, minimizing unnecessary packet forwarding.

PIM-SM relies on the concept of shared trees and SPTs to forward multicast traffic. Initially, multicast traffic follows the RPT from the RP to the receivers. However, if a receiver is sufficiently close to the source, the router can switch to a more efficient SPT directly connecting the source and the receiver, bypassing the RP. This switch to the SPT is triggered by the receiver sending a prune message towards the RP, indicating that it no longer needs to receive multicast traffic through the RP.

Figure 2.2: PIM Sparse Mode



Overall, PIM-SM optimizes multicast traffic delivery by establishing forwarding paths only when necessary, utilizing shared trees and SPTs, and efficiently switching between these paths based on receiver proximity to the source. This results in

reduced network congestion and improved scalability compared to PIM Dense Mode.

2.3.5 PIM-SM Source Specific Multicast

For a one-to-many application model, it would be more beneficial to configure for both the group and the source, thereby removing the need for the network to manage source discovery. This gave rise to SSM supporting one-to-many applications (broadcast applications) with a datagram delivery model.

(S,G) represents a specific source (S) that transmits multicast traffic to a multicast group (G). (,G) denotes any source that transmits the multicast traffic to a multicast group (G).*

In SSM, receivers use IGMPv3 to subscribe to a specific channel identified by both the source address (S) and the multicast group address (G). This tells the network the receiver only wants traffic from a specific source sending to a particular group. Unlike traditional multicast, SSM avoids flooding the network with group addresses. Instead, each source manages its own channels, and applications on the same source can use unique groups to avoid conflicts. Interestingly, different sources can reuse the same group address without causing issues, further reducing network overhead [5].

Undoubtedly, PIM-SSM mode with IGMPv3 comes with various advantages over PIM-SM. PIM-SM doesn't have network-based source discovery, due to which the application should be configured for both a group and a source, not just a group. When utilizing PIM-sparse mode, the receivers depend on the RP to identify new sources within the network. On the other hand, an IGMPv3 lets the receivers join multicast groups from specified source addresses. In addition to simply joining any group, the receiver has the ability to receive the group from a designated source. Also, when employing SSM / IGMPv3, shared trees are non-existent. Only the SPTs are built aimed at the sources, eliminating the need for RPs, Auto-RP, or

Bootstrap.

2.3.6 Bidirectional PIM

Bidirectional PIM establishes bidirectional multicast trees without source-specific information at each node. It enables two-way traffic flow, treating all sources as potential receivers, enhancing efficiency in many-to-many applications. By utilizing only shared trees $(*,G)$, it reduces resource usage and requires each router to maintain just one shared tree. Unlike PIM-SM, which necessitates both $(*,G)$ and (S,G) entries, bidirectional PIM only employs $(*, G)$ for the shared tree, allowing any source to send to the RP without PIM registration. Strategic RP placement is crucial to avoid loops, with a designated router forwarding multicast traffic to the RP based on the lowest metric or highest IP address.

2.3.7 Link state multicast routing

Link-state protocols have the capability to accommodate multicast routing by providing each router with a comprehensive network perspective, enabling the calculation of optimal source-specific trees. However, this process is resource-intensive and does not scale efficiently. MOSPF (Multicast Open Shortest Path First), outlined in and deprecated in, enhances the OSPF link-state protocol by introducing a new type of LSA while maintaining compatibility with the unicast protocol variant. It seeks to enhance performance by deferring tree computation until the detection of the first packet from a new source, thereby implementing on-demand routing. Nonetheless, these efforts fall short of achieving true scalability for the protocol.

2.3.8 Core-Based Trees and PIM-SM

- *SPT (Source Path Tree)*: SPT, denoted as (S, G) , representing the source IP address (S) and multicast group address (G) , is a multicast distribution tree

in network routing that has its root at the source of the multicast data and branches out to all receivers. The SPT ensures that multicast data is delivered via the shortest path from the source to each receiver, which makes it an optimal route for data transmission, reducing latency and network congestion [6].

- *Shared Trees (unidirectional/bidirectional)/Rendezvous Point Trees (RPT):* Contrary to SPT that originate from the source, shared trees, or RPT, or Core Based Trees (CBT) employ a single common root (rendezvous point) centered at one specific router called the core. SPTs can establish the most efficient route for multicast traffic between the source and receivers. However, this targeted approach in SSM can strain routers with more complex forwarding compared to traditional multicast. Further, each router maintains state information for every active source, which can strain resources in large networks. On the other hand, shared trees simplify router management, but the drawback is that it can lead to suboptimal paths for data, potentially increasing latency.

2.3.9 Other relevant issues

Group communications present unique challenges compared to unicast scenarios, leading to a mismatch with many applications, protocols, and conventional practices. For example, IP multicast lacks inherent security measures, as the identities and quantities of receivers are typically unknown beforehand to senders and routers. This uncertainty regarding security and scalability often deters service providers from widespread multicast deployment, unless in tightly regulated environments. Additionally, traditional IP multicast's lack of access control enables any host to join any group, causing dynamic group sizes and potential for denial of service (DoS) attacks. Security mechanisms like IPsec and SSL/TLS, tailored for point-to-point

communications, require adaptations for multicast scenarios, with varying authentication and privacy requirements depending on the application. SSM addresses some limitations but falls short in areas like IP address spoofing prevention and sender authentication. Achieving reliable multicast transmissions presents further challenges, as TCP's acknowledgment mechanism is not well-suited for multipoint scenarios, necessitating alternative approaches for reliability enforcement, either through specialized transport protocols or at the application layer.

2.3.10 Delivering multicast service using IGMP

IGMP (Internet Group Management Protocol) is a cornerstone protocol for efficient multicasting in IPv4 networks. It operates directly on top of IP, allowing devices to join or leave multicast groups designated by a specific IP address range (class D: 224.0.0.0 to 239.255.255.255). This enables them to receive the same data stream simultaneously, reducing network traffic compared to sending separate copies. This is especially important in live media production where each data flow (like video or audio) is mapped to a unique multicast group for efficient routing. Protocols SMPTE ST2110 and ST2022-6 leverage this functionality to represent individual essences and SDI stream elements, ensuring smooth and efficient production workflows.

IGMP plays a crucial role in managing multicast traffic, and its transmissions, such as "join group" or "leave group" messages, allow for dynamic changes in multicast group memberships. Additionally, IGMP oversees group subscriptions, with routers updating the Multicast Routing Information Base (MRIB) based on received or initiated receiver subscriptions for specific groups.

Compared to older versions, IGMPv3 is the most advanced and efficient version protocol. It improves upon previous versions by introducing source-specific multicast, which allows hosts to specify the specific sources they want to receive multicast

traffic from. IGMPv3 also includes a leave group mechanism, which helps to reduce latency when hosts need to stop receiving multicast traffic. Additionally, IGMPv3 maintains backward compatibility with older IGMP versions, so it can be used on networks that are not yet fully upgraded to IGMPv3.

2.3.11 IGMP Snooping

IGMP presents challenges for switches at the data link layer by potentially flooding multicast traffic to all devices, causing network slowdowns. IGMP snooping addresses this issue by allowing switches to selectively forward multicast traffic only to interested devices, enhancing network efficiency. Originally, layer 2 switches broadcast multicast frames to all ports, overwhelming devices not expecting such traffic. IGMP snooping dynamically populates MAC tables, limiting multicast traffic to necessary locations. However, implementation details vary as IGMP snooping is hardware-dependent. IGMPv3 compatibility adds complexity, requiring switches to process joins like IGMPv1/v2 and track member tables with include/exclude lists for each source. In essence, layer 2 switches monitor IGMP and PIM packets to build the multicast MAC table, identifying router and member ports to prevent flooding. Implementing IGMP snooping, especially for IGMPv3, can be complex, so it's advisable to proceed with caution to avoid disrupting network operations.

2.3.12 Layer 3 vs Layer 2 multicast

Layer 3 multicast traffic is managed by IP multicast routing protocols such as PIM, enabling routers to efficiently forward multicast traffic across the network. When a source sends multicast traffic to a multicast IP address, routers use protocols like PIM to determine the optimal path for delivering the traffic to all interested receivers. This approach allows for scalable and efficient distribution of multicast traffic across large networks, as routers base forwarding decisions on the multicast

group memberships of receivers.

Layer 2 multicast traffic management involves protocols like IGMP and Ethernet MAC addresses. Switches utilize multicast MAC address mapping to selectively forward multicast traffic to required network segments (VLANs) instead of flooding all ports like broadcast traffic. Devices interested in receiving multicast traffic send IGMP reports to local routers, which then forward multicast traffic only to segments with interested receivers, optimizing network efficiency.

2.3.13 STP Protocol

Spanning Tree Protocol (STP) mainly prevents layer 2 loops in Ethernet networks by electing a root bridge and calculating the shortest paths to it from each switch. Redundant paths are then identified and blocked by placing certain switch ports in a blocking state, thus preventing loops from forming. STP continuously monitors the network topology for changes and reconfigures as needed to maintain a loop-free topology. This loop prevention mechanism ensures stable and reliable network operation, preventing broadcast storms and network outages.

2.4 Time management protocols

In multicast, maintaining precise clock synchronization across receivers is vital for seamless audio/video stream delivery and avoiding playback inconsistencies.

- Network Time Protocol (NTP) : NTP acts as the network's metronome, ensuring all receivers experience content in unison, eliminating jitter and lag. This synchronization is achieved through timestamps embedded in the SDP that reference NTP time, enabling a unified playback experience for all participants regardless of their network location. Essentially, NTP ensures everyone joins the virtual orchestra at the same downbeat, resulting in a harmonious

and immersive multimedia experience.

- Session Description Protocol (SDP) : SDP acts as a session's digital passport, allowing receivers to easily find and join multimedia gatherings. Instead of individual invitations, SDP broadcasts essential details like timing, media types, and transport information, simplifying participation and ensuring everyone's on the same page. Often paired with Session Initiation Protocol (SIP) for setting the stage, SDP empowers efficient and seamless multicast experiences.

2.5 Security issues in traditional IP multicast

This section delineates network vulnerabilities and corresponding mitigation strategies in the context of Traditional IP Multicast.

2.5.1 Threats against Routers

A multitude of basic threats against a router exist, regardless of its support for multicast or if the attack involves multicast traffic or protocols. Multicast attacks, often unintentional, can have significant impacts, as seen in the Witty worm incident in 2004 [7]. This worm randomly attacked IP addresses, affecting multicast IP destinations and leading to router collapses in many organizations due to overload. General threats to routers include:

- Malicious actors can launch packet flood attacks against various network components. These attacks overwhelm the target with a high volume of data packets, aiming to disrupt functionality. Targets can include:
 - Hardware paths: Slow paths (also known as punt paths) designed to handle low-priority traffic.

- Software paths: Specific network management or control plane ports used for protocols like SSH, Telnet, BGP, OSPF, and NTP. Flooding these ports can prevent legitimate communication and network management tasks.
- Router Vulnerabilities: Attackers can exploit weaknesses in router firmware or misconfigured settings to gain unauthorized access. This can allow them to manipulate network traffic, steal data, or launch further attacks. Common entry points include weak Telnet or SSH passwords and insecure Simple Network Management Protocol (SNMP) community strings.
- Operational Security Gaps: Misconfigurations in router settings or even insider attacks can compromise the entire network's security. Accidental mistakes or intentional tampering can expose sensitive data or disrupt critical network operations.

2.5.2 Network Attacks on IGMPv3

As discussed previously, IGMPv3 provides support for SSM, allowing hosts to specify the multicast sources they want to receive traffic from, in addition to the multicast groups they wish to join. Like any networking protocol, IGMPv3 is susceptible to various attacks. Here are some possible attacks on IGMPv3 and corresponding mitigation strategies:

- IGMP Spoofing: Attackers may spoof IGMPv3 messages to join or leave multicast groups, causing disruption or unauthorized access to multicast traffic. In IGMPv3, this could involve falsely claiming membership to specific multicast groups or source-filtering information.

Mitigation: Employ cryptographic mechanisms, such as message authentication codes (MACs), to authenticate IGMPv3 messages exchanged between

hosts and routers. Implement ingress filtering to validate the source and integrity of incoming IGMPv3 messages.

- Denial of Service (DoS) Attacks: Attackers can flood the network with bogus IGMPv3 messages, overwhelming routers and consuming network resources. This could lead to network congestion, degradation of service, or disruption of legitimate multicast traffic.

Mitigation: Deploy rate-limiting mechanisms to control the rate of incoming IGMPv3 messages and prevent floods of bogus IGMPv3 messages from overwhelming routers. Configure Access Control List (ACL) to filter out suspicious or malicious IGMPv3 messages before they reach the routing infrastructure.

- IGMPv3 State Exhaustion: Attackers may exhaust router resources by establishing a large number of unnecessary or malicious IGMPv3 group memberships or by causing routers to maintain excessive IGMPv3 state information.

Mitigation: Implement rate limiting and access control measures to restrict the establishment of IGMPv3 group memberships and the creation of IGMPv3 state information. Configure routers to prune unnecessary IGMPv3 state information and maintain efficient IGMPv3 group management.

- Information Leakage: Attackers may eavesdrop on IGMPv3 messages exchanged between hosts and routers to gain insights into the multicast group memberships and source-filtering information of network users, potentially compromising privacy or security.

Mitigation: Encrypt IGMPv3 messages exchanged between hosts and routers using secure communication protocols, such as IPsec, to protect the confidentiality and integrity of IGMPv3 messages and prevent information leakage.

2.5.3 Network Attacks on PIM

PIM is a routing protocol used in multicast communication scenarios. It allows routers to dynamically route multicast traffic efficiently within a network. Like any network protocol, PIM is susceptible to various attacks. Here are some common attacks and mitigations:

PIM attacks target routing disruption and information manipulation:

- **Spoofing:** Attackers forge PIM neighbor or prune messages to disrupt communication or prevent specific receivers from getting multicast data.
- **Flooding:** Malicious actors can overwhelm the network with PIM messages, causing denial-of-service.
- **Manipulation:** PIM routing information can be tampered with to redirect multicast traffic or disrupt communication.

PIM mitigations focus on securing communication and filtering messages:

- **Authentication:** Techniques like MD5 or SHA-based authentication ensure only authorized routers participate.
- **Rate Limiting:** Stops DoS attacks by limiting PIM message volume.
- **Message Filtering:** Routers discard suspicious or malformed messages based on pre-defined rules.
- **PIM Hello Timers:** Fine-tuned timers help detect and mitigate neighbor spoofing.
- **Traffic Engineering:** Optimizes multicast paths to minimize route manipulation impact.
- **Monitoring & Alerting:** Detects abnormal PIM behavior for prompt action.

- Encryption: Encapsulating PIM messages in secure tunnels protects against eavesdropping and tampering.
- ACL : Restrict which routers can send/receive PIM messages, limiting the attack surface.

2.5.4 Other drawbacks

- Slow convergence: In large networks, updating the routing tables across all routers that are involved in the multicast group can be a slow process. Smooth transitions between group membership and timely data delivery are disrupted by this delay.
- Scalability issues: Managing the state information and control messages can become complex and resource-intensive on routers when the number of receivers and sources are increased.

For such routing protocol limitations, SDN-based multicast offers a centralized controller, akin to a conductor orchestrating a symphony. This controller dynamically manages network updates, allowing receivers to join or leave groups seamlessly while ensuring efficient traffic flow, even in large, complex networks. This improved scalability allows SDN to handle vast networks and many participants with ease, addressing challenges that often cause traditional approaches to struggle.

2.6 Overview of SDN Based IP Multicast

SDN-based multicast offers a compelling alternative to traditional IP multicast (IGMP/PIM-SSM) for handling large-scale deployments. While IGMP/PIM-SSM offer some advantages, they struggle with scalability, network failures, lack of broadcast tallies, path diversity, and unpredictable behavior. Additionally, limited control

over traffic, lack of dynamic reconfiguration, and manual management increase complexity. Finally, security mechanisms might be inadequate. SDN-based multicast addresses these limitations with a centralized controller, enabling efficient traffic flow, dynamic updates, and improved security.

This section delineates the distinctive attributes of an SDN-enabled IP multicast system and elucidates their enhanced security and preference over conventional multicast approaches.

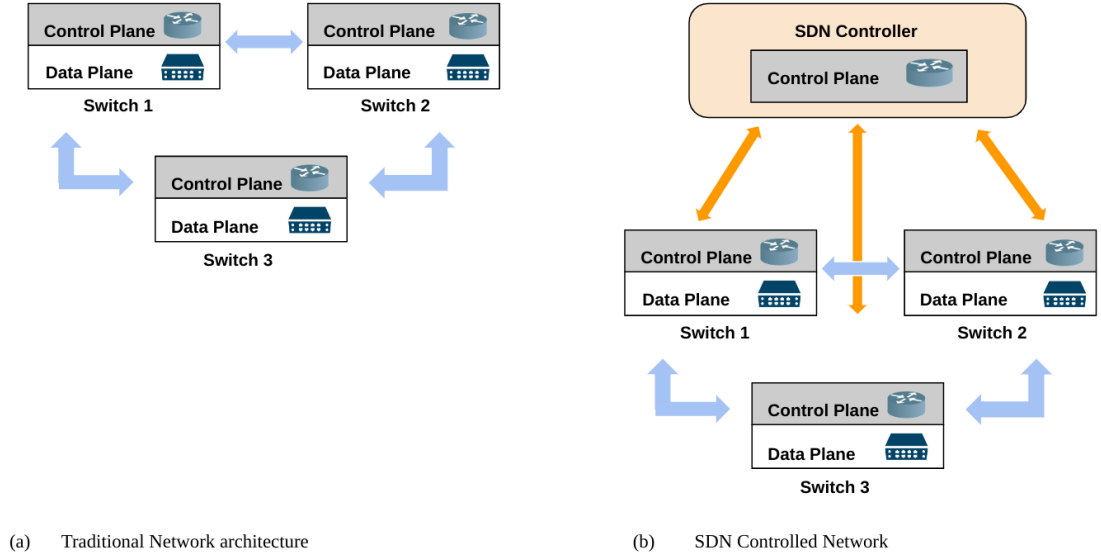
2.6.1 Software-Defined Networking

Traditional networks (often referred to as SDI-based) rely on physical infrastructure. Dedicated hardware like routers, switches, and firewalls manage and direct traffic flow. These devices utilize standardized protocols like TCP/IP and Ethernet for communication over physical cables. This approach offers stability but can be less flexible and more complex to manage compared to newer software-defined networking solutions. However, they are primarily designed for manual, individual device management, lacking centralized control, programming, or automation. Consequently, rapid network reconfigurations demanded by modern enterprises and data centers often exceed manual processes' capabilities. For instance, adding or moving a device entails configuring multiple network devices and updating various network settings, which can be time-consuming, error-prone, and unscalable. Furthermore, using appliances from different manufacturers can lead to incompatibility issues due to variations in protocol implementation, leading to vendor lock-in.

Unlike traditional networks, Software-Defined Networking (SDN) leverages software applications and APIs to control the underlying hardware. This separation allows for a more dynamic and programmable approach to network management. Network administrators can centrally configure and optimize traffic flow, fostering greater flexibility and innovation compared to static, hardware-driven networking. The

key difference between SDN and traditional networking lies in their core infrastructure; SDN is software-centric, while traditional networking is primarily hardware-dependent.

Figure 2.3: Traditional Networking vs Software Defined Networking



Benefits of SDN

In traditional networking, devices like routers have two main components; control plane and the data plane. *Control Plane*, is referred as the network's "brain" that dictates the data traffic route and instructs the *data Plane*, which acts on these instructions by forwarding or blocking data packets on specific interfaces. Each device in a legacy network has its own control and data planes, making traditional networks characterized by a decentralized control plane. Therefore, configuring or making minor modifications that require individual access to each device can be usually expensive.

The distinction between the control and data planes is not new. Modern multi-slot routers or switches have dedicated processors for the control plane and independent line cards for data plane switching functions. The control plane, running on

a dedicated processor, manages the forwarding tables for inbound-outbound interface switching. The data plane, implemented on an independent line card, performs switching functions. Control plane messages or unknown packets are sent to the route processor for further processing. Such architecture is optimized using MPLS protocol which carries control traffic through a dedicated route processor while using a label-based switching paradigm on a different line card for high performance (shown in Figure 2.3(a)).

Before the idea of SDN came along, control and data planes were closely connected and managed together, almost like a finely-tuned orchestra. This setup led to the creation of specialized network elements that could handle different tasks. But, while this approach did offer high performance, it was also complex and expensive. This is one of the reasons why SDN, which separates the control and data planes, is seen as a good alternative. Using an SDN controller simplifies the process as network management is centralized to a single console. The controller communicates with networking devices like switches and routers. This eliminates the need to individually manage each networking device in a purely SDN-enabled network. This approach stands in contrast to traditional networks where dedicated hardware like routers and switches perform all traffic management tasks. SDN offers a more software-centric approach, providing greater control and flexibility. SDN can form and oversee a virtual network or control traditional hardware through software [8].

Decoupling: SDN separates the data and control planes and implements the data plane in software, providing programmatic access and significantly enhancing network administration flexibility. This decoupling means the two planes operate independently and are oblivious to each other, enabling dynamic access and management. SDN empowers network administrators to manage and configure traffic flow from a single, centralized control console. This eliminates the need for manual adjustments on individual switches, simplifying network management and enabling

quicker responses to changing network demands. This allows the administrator to modify any network switch's rules as necessary, providing a highly detailed level of control over prioritizing, deprioritizing, or even blocking certain types of packets (shown in Figure 2.3(b)). Moreover, decoupling simplifies the process of troubleshooting network architecture. It also aids in the development of a more scalable and adaptable infrastructure that can more readily adapt to changing business needs. Since the control plane of SDN is based on software, it offers more flexibility than traditional networking, empowering administrators to manage the network, modify configuration settings, allocate resources, and enhance network capacity - all through a centralized user interface, eliminating the need for additional hardware.

2.7 SDN Architecture

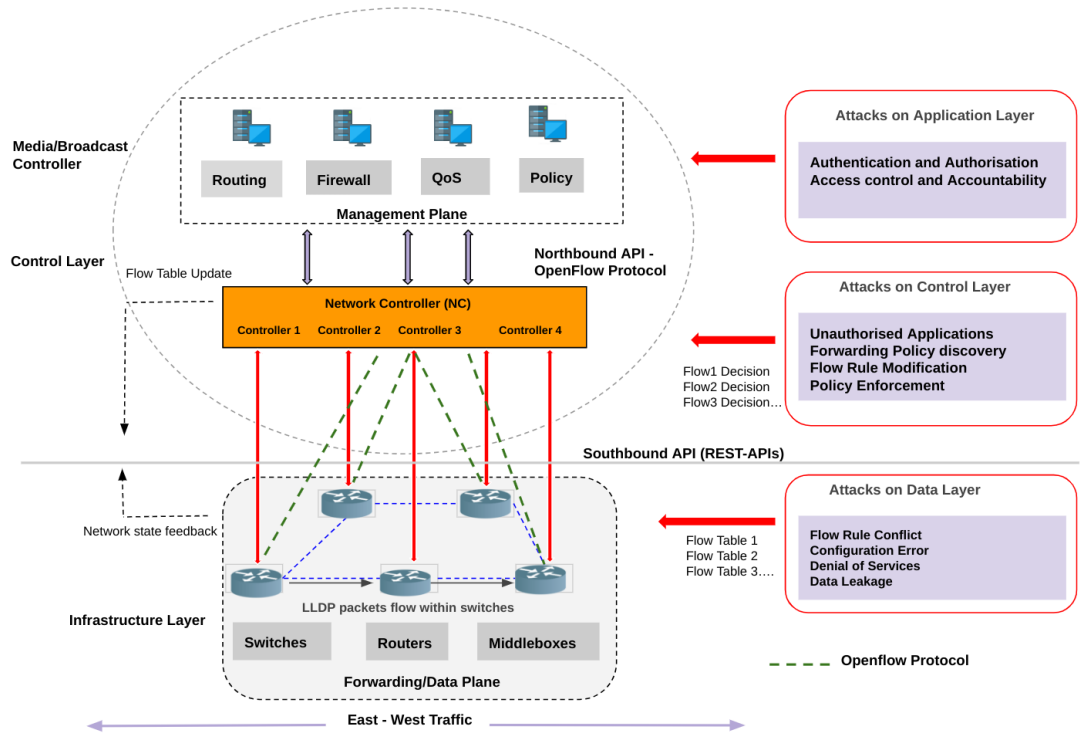
In traditional networks, each switch maintains separate data and control planes. The control planes on these switches communicate with each other to share network topology information. This information is used to build forwarding tables within each switch's data plane. These forwarding tables ultimately determine the path incoming data packets take through the network. SDN breaks away from the traditional distributed approach by centralizing the control plane in a dedicated software component called the SDN controller (Figure 2.4). This central controller empowers network administrators to manage and configure traffic flow from a single console, eliminating the need for manual configuration on individual switches.

The data plane remains within the switches. When a packet arrives, the switch consults its flow table – a set of entries pre-programmed by the controller – to determine the packet's forwarding behavior. Each flow table entry contains match fields (like input port number and packet header information) and corresponding instructions. The switch first matches the packet against the match fields in the flow table. If there's a match, the switch executes the instructions associated with that entry,

which might involve forwarding the packet to specific ports, discarding it, or even modifying its header.

In cases where the packet doesn't match any existing flow table entry, the switch can consult the SDN controller [8]. The controller then generates a new flow entry and sends it to the switch, enabling the switch to forward or discard the packet accordingly.

Figure 2.4: SDN with Openflow controller Architecture



SDN architecture consists of three layers: the application layer, control layer, and infrastructure layer (Figure 2.4). Applications like intrusion detection and firewalls reside in the *application layer*. The *control layer* acts as the brain with the SDN controller, providing an abstraction layer for applications. Finally, the *infrastructure layer* comprises physical switches that handle actual data packet movement.

These layers interact through northbound APIs (applications to control layer) and southbound APIs (control layer to infrastructure layer).

2.7.1 Overview of SDN security features

SDN, powered by OpenFlow, injects programmability, dynamism, flexibility, and intelligence into network architectures. These benefits stem from four key features:

(i) Real-time Traffic Manipulation : SDN's core principle of consulting the control plane when the data plane lacks a matching flow rule empowers network applications with dynamic flow control. This capability shines in applications like dynamic load balancing and network management tools, allowing them to adjust traffic routing in real-time based on network conditions.

(ii) Centralized Network Overview and Control : In SDN, all network devices (data plane) connect to a central control plane. This central hub distributes control messages, like flow rules and configuration updates, to the devices. Additionally, the control plane actively gathers network status information from each device. This centralized view and control capability empower network applications running on the control plane to manage all devices and monitor the entire network.

(iii) Programmable Network Behavior : SDN's programmability unlocks the potential to design entirely new network functions. Similar to how apps extend a smartphone's capabilities, network applications running on the control plane can introduce innovative functionalities. To facilitate this, various network programming languages have emerged, making it easier to write programs that control and manage the network.

(iv) Streamlined Data Forwarding : A core principle of SDN is the separation of the control plane, responsible for network intelligence, from the data plane, focused on efficient packet forwarding. This simplifies the data plane logic, making it more efficient and potentially adaptable.

This section describes how network security benefits from each SDN feature. State-

Table 2.1: SDN for Enhanced Security: Exploring Control and Visibility Features

SDN Features	Description	Security Benefits	Network/Security Applications	Defense Roles
Real-time Traffic Manipulation	Dynamic control of the network by rerouting, forwarding, dropping packets.	Filter network traffic in real-time to isolate malicious packets from normal flows.	FlowVisor, NVP, OpenFlow, Random route mutation	Prevention, Response
Centralized Network Overview and Control	The controller monitors and manages all network flow and status information.	SDN's centralized monitoring unlocks comprehensive security. Visibility across the network streamlines anomaly and flood detection.	DDoS detection/defense, OpenFlow-Tags, Cloud-Watcher	Detection, Response
Programmable Network Behavior	SDN allows for the creation of custom network functions via programming.	SDN simplifies security app development, fostering innovative solutions.	Frenetic, OpenSAFE, Controller Programming	Detection, Response
Streamlined Data Forwarding	SDN centralizes intelligence, freeing data plane devices for fast packet forwarding.	SDN integrates modular security apps, enabling lightweight data plane adjustments for a dynamic security shield.	OpenSDWN	Prevention, Detection, Response

of-the-art research examples showcase security applications leveraging these features. Table 2.1 summarizes this knowledge. This analysis dives into how SDN empowers network security. We'll dissect core SDN functionalities, explore how each strengthens security posture, and showcase practical applications. Finally, we'll examine how these features can fit within the layered defense-in-depth security framework, providing a comprehensive overview of SDN's contributions to network protection.

2.7.2 Benefits of Network security

A. Real-time Traffic Manipulation

SDN's dynamic flow control unlocks a range of new network security possibilities. One example is dynamic access control, traditionally implemented through independent middleboxes like firewalls. With SDN, network devices like OpenFlow switches/routers can handle access control directly, eliminating the need for additional hardware. Furthermore, SDN allows for granular control of network flows (from single packets to full flow entries), enabling more efficient traffic management.

Another advantage is the ability to dynamically separate malicious or suspicious traffic from legitimate flows. This is valuable for differentiating security services. Imagine using a Network Intrusion Detection System (NIDS) to monitor traffic and identify potentially malicious flows. Traditionally, further investigation might involve rerouting suspicious traffic through a proxy server for deeper inspection with security tools like honeypots.

B. Centralized Network Overview and Control

Monitoring an entire network is crucial for maintaining security, but traditional methods of tracking subnet activity—whether internal or through traffic—often require deploying sensors and gathering information from various network devices, which can be challenging in large-scale environments. SDN offers a more efficient approach, enabling centralized control and monitoring across the network.

SDN's centralized architecture allows for easy monitoring of network devices by collecting statistics and processing flow requests from each device. This holistic view aids in detecting and defending against network-wide attacks. Administrators can use anomaly detection techniques to identify unusual patterns and adapt the network's resources to counteract large-scale attacks.

Additionally, SDN improves the utilization of security devices by rerouting specific network flows through necessary security appliances, like hardware-based middle-

boxes and virtual network functions. In complex network setups, ensuring that most network traffic passes through security devices can be difficult because of their fixed physical locations. SDN's network-wide visibility addresses this by providing a clear understanding of traffic patterns and security appliance locations, enabling administrators to reroute traffic as needed to ensure it goes through the appropriate security devices. This flexibility can enhance the effectiveness of security measures and make large-scale network monitoring more manageable.

C. Programmable Network Behavior

Traditional network security relies on pre-configured hardware middleboxes or software programs. These solutions offer limited flexibility, as modifying their functionality can be difficult. Additionally, unforeseen security needs can leave deployed solutions inadequate. Replacing them often requires purchasing new hardware, leading to increased costs.

SDN's network programmability offers a compelling alternative. It empowers the creation of custom network security applications directly on the control plane. This eliminates the need for additional hardware or software purchases. For instance, network scan detection applications or even intelligent solutions like DDoS detection can be programmed within the SDN environment. This approach is not only cost-effective but also allows for dynamic security adaptations as network requirements evolve.

D. Streamlined Data Forwarding

Unlike traditional network devices, SDN simplifies the data plane hardware. This is achieved by separating complex control logic functions into the centralized controller. This streamlined data plane, composed of relatively simple modules, becomes more adaptable. New functionalities, including security features, can be readily extended through software modifications. This enables the data plane to play a more active role in network security.

2.7.3 SDN: Revolutionizing Information Security

We've explored how SDN features enhance network security. Now, let's see how they support the core information security triad: prevention, detection, and response.

A. Prevention

Prevention involves blocking attackers from accessing critical network resources through robust security policies. This requires careful planning to prevent misconfigurations that might either block legitimate users or allow malicious access. Traditional static policies lack the flexibility for complex, evolving networks, leading to a rise in dynamic access control. This dynamic approach typically involves integrating middleboxes into network infrastructure for added security.

SDN streamlines prevention with its dynamic flow control feature, allowing access control without additional middleboxes. SDN's centralized controller can adapt security policies in real-time, automatically updating when the data plane requests instructions for a network flow. However, dynamic flow control introduces potential security risks, such as dynamic flow tunneling, where load balancers may alter packet headers, allowing malicious traffic to bypass access controls. This vulnerability has been addressed in some SDN controllers like Floodlight and NOX but not all. Users of dynamic access control must be vigilant to avoid these risks.

B. Detection

Detection is crucial for identifying network intrusions, with two primary methods: misuse detection, recognizing attacks based on known patterns, and anomaly detection, spotting deviations from expected traffic behavior. SDN's centralized control and network-wide visibility enhance anomaly detection, providing a comprehensive perspective for detecting unusual or malicious activity. Centralized control is especially useful in detecting distributed attacks like flooding, where traffic from multiple sources can be quickly identified.

SDN/OpenFlow also supports misuse detection by providing network information,

including packet payloads. While it's impractical to send every packet to the control plane, Network Intrusion Detection Systems (NIDS) can leverage SDN features to inspect suspicious packets. If a new packet seems suspicious, the control plane can mirror it for deeper inspection, allowing for the detection of known attack patterns and generating alerts when necessary. SDN's programmability and network-wide visibility support the development of advanced distributed intrusion detection systems, making it easier to safeguard modern networks against various threats.

C. Response

Response to attacks is a critical aspect of network security, historically challenging due to complex middlebox configurations for dropping malicious traffic or isolating compromised hosts. SDN's dynamic flow control simplifies this process by enabling rapid reconfiguration. Attacks can be dropped, and compromised hosts can be quarantined or isolated with minimal effort, reducing the risk of attack spread.

SDN's programmability offers further flexibility, allowing response actions like network isolation and quarantine to be implemented with simple configurations. Tools like FRESCO can achieve quarantine or isolation of infected hosts with just a few lines of script, transforming a traditionally complex task into a manageable one. This dynamic reconfiguration capability improves response times, streamlines mitigation, and enhances the overall security posture of the network.

2.8 Seamless Switching for Media Streams: SMPTE 2022-7

SMPTE ST 2022 standards encapsulate video data into RTP/UDP/IP streams, facilitating seamless interoperability across SD and HD video equipment from different manufacturers. These standards address the challenge of diverse media content transmission over IP while managing errors. ST2022-7, in particular, enables

seamless protection switching of RTP datagrams, vital for broadcast workflows to maintain uninterrupted audio and video playback despite packet loss. It establishes conditions for redundant data streams, allowing receivers to switch seamlessly at the packet level without compromising data content or stream stability, often implemented through red and blue networks to ensure redundancy and reliability. Additionally, it offers flexibility for various data types and applications beyond broadcasting.

2.8.1 Red and Blue networks

In unicast streaming where each client connects to the server individually uses HTTP that has a retransmission mechanism with a latency of 20-30 seconds.

On the other hand, multicast streaming often utilizes RTP over UDP (User Datagram Protocol). RTP is a protocol used for delivering audio and video over IP networks, and UDP is the transport protocol commonly used with it. UDP is preferred for real-time streaming applications like multicast because it offers lower overhead (with latency of 125-600 milliseconds) and less delay compared to TCP (latency 20-30 seconds), which is used in HTTP. However, UDP lacks built-in error correction and re-transmission mechanisms.

Due to the lack of re-transmission mechanisms in RTP/UDP for lost or corrupted packets, the automatic recovery poses a challenge, as missing packets are not typically re-transmitted to individual receivers. To address this, redundant networks, such as "red" and "blue" networks, are utilized to mitigate packet loss risks. These redundant pathways offer duplicate routes for data transmission, reducing the likelihood of packet loss due to network failures or congestion, ensuring more reliable multicast streaming delivery.

A and B networks follow a similar principle of redundancy. In this setup, there are two separate network paths, often labelled as "A" and "B." The primary network

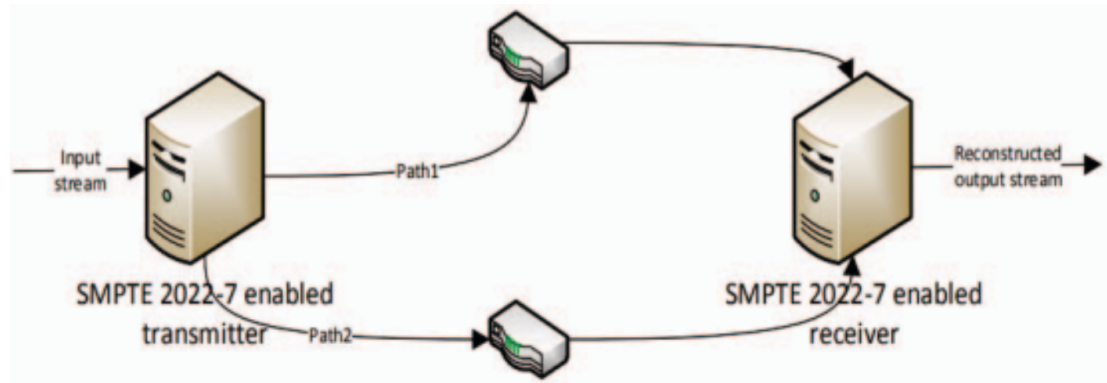
path (A) carries the normal data traffic, while the secondary network path (B) serves as a backup. If the primary network fails or experiences issues, the system switches to the secondary network to maintain connectivity and service.

This redundancy helps maintain the reliability and quality of the multicast stream, especially in critical applications like live broadcasting or teleconferencing.

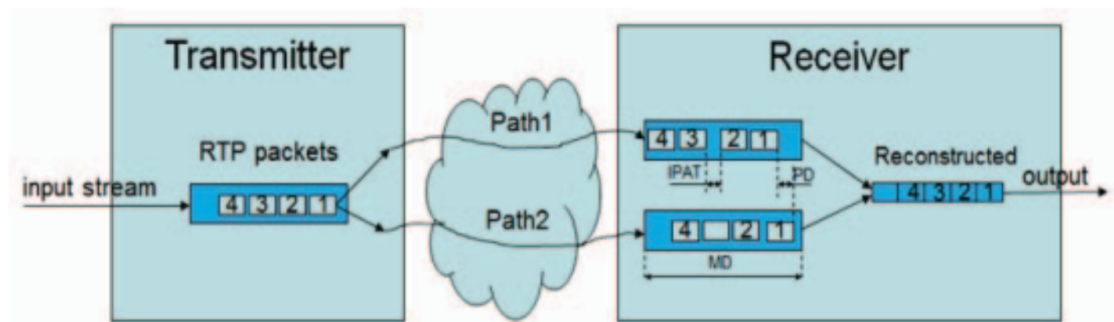
2.8.2 Functioning of redundant IP packets

SMPTE ST 2022-7 establishes two redundant RTP media networks, "red" and "blue," ensuring seamless playback through redundant packet delivery. A short buffer in the receiver stores packets from both streams, allowing playback to continue uninterrupted if a packet is lost on either path 1 or path 2 as shown in Figure 2.5 (a) [9]. This redundancy is facilitated by RTP, which timestamps and sequences each UDP packet, enabling the receiver to identify and replace missing packets. By transmitting identical program copies along different paths, ST 2022-7 mitigates single points of failure, ensuring uninterrupted workflow even amidst network path losses or intermittent packet losses on either path, provided redundancy is maintained. Buffering at the receiver compensates for network jitter, enabling seamless switching between packets from path 1 and path 2 without disrupting the output stream 2.5 (b)[9].

Figure 2.5: SMPTE 2022-7 Standard [9]



(a) Transmission between two identical IP packets



(b) SMPTE-2022-7 protocol

SMPTE ST-2110 offers following architectures:

- Spline/Monolithic: Suitable for space-limited environments with high multi-cast demand e.g. OB trucks.
- Spine/Leaf with Red/Amber and Blue networks: It offers varied bandwidth allocation for audio, ancillary data, and video signals, making it suitable for diverse multimedia applications. This architecture operates within a two-layer framework, featuring a full mesh topology that facilitates direct connections between spine and leaf switches. By bypassing hierarchical constraints, Spine/Leaf networks ensure efficient communication and data transfer, enabling seamless integration and scalability in modern networking environments.

2.9 Multicast Support in EVPN-VXLAN

Multicast support in EVPN-VXLAN overlay networks enables efficient handling of multicast traffic within Ethernet VPNs (EVPN) deployed over Virtual Extensible LAN (VXLAN) technology. Unlike traditional IP multicast networks, multicast traffic in EVPN-VXLAN is encapsulated within VXLAN tunnels and forwarded across the underlay network using unicast communication. Leveraging ingress replication, multicast traffic is sent as unicast from the source to egress routers (leaf switches), which then replicate it to interested receivers within their VXLAN segment. EVPN provides the control plane, facilitating efficient distribution of multicast group membership information using BGP, thus enabling dynamic provisioning and scaling of multicast services. This approach simplifies network design and improves scalability, making EVPN-VXLAN ideal for modern data center and service provider environments.

Multicast support in EVPN-VXLAN overlay networks operates within the broader context of SDN-controlled networks. While EVPN-VXLAN focuses on efficiently handling multicast traffic within the overlay technology stack, SDN encompasses a broader set of principles for network management and control, including but not limited to multicast support [10].

2.10 Conclusion

In conclusion, multicast technology offers a sophisticated approach to data transmission, particularly beneficial in scenarios where content needs to be efficiently distributed to multiple recipients simultaneously. Unlike unicast and broadcast methods, multicast optimizes network bandwidth usage by transmitting data from a single source to multicast group which is a specific group of recipients. This targeted approach minimizes network congestion and enhances overall efficiency, making it

ideal for applications like streaming media and online gaming.

Multicast can operate at both the network layer, if supported by routers, or as an application-level overlay network, leveraging regular hosts for routing and switching. Various multicast routing protocols, such as dense mode and sparse mode, facilitate the dissemination of data to multiple recipients. DM protocols flood the network with multicast traffic, assuming every subnet has recipients, while SM protocols operate on a pull mechanism, delivering traffic only upon request. The implementation of multicast involves careful consideration of routing algorithms, such as Source-Specific Trees and Shared Trees, to optimize data transmission paths. Bidirectional PIM and Source Specific Multicast (SSM) further enhance efficiency in many-to-many applications by allowing two-way traffic flow and eliminating the need for source discovery, respectively.

Despite its advantages, multicast technology poses challenges, including security concerns, scalability issues, and the need for specialized protocols and mechanisms to ensure reliable transmissions. Traditional IP multicast protocols such as IGMPv3 and PIM have proven vulnerabilities to various network attacks, including IGMP spoofing, DoS attacks, state exhaustion, and information leakage. Mitigating these attacks requires the implementation of cryptographic mechanisms, rate-limiting, message filtering, and access control measures. However, these traditional protocols still present limitations, especially in handling multicast signals, network failures, and security.

The emergence of SDN offers a promising solution to address the shortcomings of traditional multicast approaches. In SDN architecture, control plane and data planes are separated by SDN controller, providing centralized control and programmability, which enhances network management flexibility and scalability. By utilizing SDN-enabled IP multicast systems, broadcasters can achieve enhanced security, multicast addressing efficiency, and dynamic flow orchestration. Moreover, advancements such

as SMPTE ST 2022-7 for seamless protection switching, SDN-based architectures with Red/Blue Spine-Leaf networks, and multicast flow orchestration solutions like Arista EOS Media Control Services (MCS) further improve the reliability, scalability, and performance of IP multicast workflows. Policy-based traffic management mechanisms such as SR-TE policy and Policy-Based Routing (PBR) enhance network efficiency and resource utilization.

In essence, transitioning from traditional IP multicast to SDN-based multicast architectures empowers broadcasters to transcend existing limitations and attain heightened levels of security, scalability, and flexibility while efficiently managing multicast traffic in contemporary broadcast environments. These advancements facilitate seamless live media production workflows, ensuring the reliable delivery of high-quality content to audiences. SDN-enabled IP multicast achieves enhanced security through centralized control, granular access control, dynamic adaptability, end-to-end encryption, and improved visibility, thereby enforcing uniform policies, thwarting unauthorized access, swiftly responding to threats, encrypting data transmission, and proactively detecting and mitigating risks, thus surpassing the security capabilities of traditional multicast systems.

3 Cyberattacks in SDN

While SDN-based IP multicast systems offer flexibility and efficiency, they also introduce new attack vectors. To secure SDN, strong authentication must be employed, TLS/SSH encryption, and regular updates for the controller. Role based ACL must be used and flow verification to restrict switch access, ensure redundant communication paths, and deploy DoS mitigation like rate limiting and traffic filtering. Switch firmware must be updated regularly, hardware security must be utilised, and real-time monitoring for swift threat detection and response must be employed.

Kreutz et al. [11] highlight the security challenges in SDNs, revealing a lack of reliable mechanisms to establish trust between applications and the controller. This void can allow malicious users to inject deceptive traffic flows into controllers and switches, potentially generating a DoS attack using network elements like servers, switches, and computers. Attacks on controller and switch vulnerabilities can significantly disrupt the network, with malicious controllers compromising the entire network. In their research, Abdulkarem and Dawod (2020) investigated the use of Python to develop a DDoS attack mitigation and detection solution specifically targeting the data layer within an SDN environment [12]. Their work sheds light on the potential for SDN-based approaches to combat DDoS attacks at this critical network layer. Security vulnerabilities in the SDN controller, such as weak authentication, information disclosure, and incomplete encryption, can pose a significant threat to the entire network [13]. If not adequately secured, a compromised controller could

disrupt network operations or be exploited to launch further attacks. This emphasizes the importance of prioritizing robust security measures for SDN controllers. Switches, with their relatively low hardware performance, are susceptible to attacks on communication channels aiming to break the link between switches and controllers. OpenFlow allows SDN switches to operate independently (standalone) or drop traffic (fail-secure) when disconnected from the controller [14]. Implementing multiple controllers can fragment the network, potentially leading to consistency and privacy concerns [15].

Maintaining the integrity and validity of flow rules within the data plane remains a critical security concern in SDN architectures. Inconsistency issues can arise from malicious tampering or transmission delays during the release process between switches and controllers [16][17].

Following are some possible attacks on SDN-based IP multicast systems and corresponding mitigation strategies in detail:

3.1 Attacks on Infrastructure Plane

In SDN architectures, applications are situated atop the control plane, exposing it to potential security threats. The integration of third-party applications makes securing the controller particularly challenging in SDN architecture. Thus, malicious applications can significantly impact the network similarly to a compromised controller [18]. Applications deployed within the SDN architecture can provide additional security risks if they contain vulnerabilities or design flaws. These vulnerabilities could be weaponized by attackers to breach security and compromise network functionality [15].

- Malevolent OpenFlow(OF) switches : Redirecting the flow of network through a compromised OF switch that enables to manipulate the IP packets, resulting

in diversion of the flow of the network and dropping of reliable traffic, thereby disrupting communication among SDN devices. While extended idle duration can be a symptom of the manipulation, it's not necessarily the direct cause of network congestion. The manipulation itself might be creating excessive traffic or redirecting traffic in unintended ways, leading to congestion. This can consequently overwhelm the controller, leading to packets dropping or flooding unnecessary "Packet_In" messages from overloaded switches [19].

- Malicious hosts in the infrastructure plane : Malicious hosts possess the capability to launch attacks on any switch or controller within an SDN infrastructure by producing falsified network packets. These forged packets can have various fields altered, such as the IP or MAC fields, in an attempt to obfuscate the attacker's identity. Additionally, malicious hosts can initiate DoS attacks by inundating OF switches with millions of packets, aiming to overwhelm their memory capacity. Moreover, each new forged packet, indicated by a unique source IP address, prompts the OF switches to send redundant Packet_In messages to the controller, potentially impairing the controller's performance [19].

3.2 Attacks on Control Plane

Attackers increasingly target the control plane, exploiting its critical role in network management, abstraction, and application support.

- Link Flooding Attack (LFA) : LFA floods the connection to the destination server to induce overcrowding, thereby disrupting legitimate traffic. The attacker first identifies the target server by sending trace routes and then generates a map of links surrounding it.
- Topology Discovery Attack : Using the OpenFlow Discovery Protocol (OFDP),

the SDN controller gathers network topology information. However, attackers can manipulate network topology status by sending unauthorized Link Layer Discovery Protocol (LLDP) packets lacking authentication messages [20].

- Man in the middle (MITM) attack : This attack involves a malicious actor acting as a middleman between two parties. They secretly listen in on, or even alter, the communication passing between them. Techniques used for this include DNS spoofing, session hijacking, and port monitoring. Attackers employ MITM attacks to steal login credentials or personal information, engage in espionage by spying on victims' communications, sabotage ongoing communication channels, or corrupt data exchanged between controllers and switches. This attack is a standard method of infiltrating networks, involving the insertion of an agent node between the source and destination nodes to intercept and manipulate transmission data without detection by either communicating party.
- Inserting fraudulent flow rules : A core challenge for SDN lies in securing the controller, the brain of the network. While it efficiently translates application needs into flow rules, malicious actors could exploit vulnerabilities to inject unauthorized rules, potentially compromising network security.

3.3 Attacks on Application Plane

In the SDN application plane, modules like load balancing, routing, firewall, and intrusion detection execute tasks such as traffic monitoring, analysis, authentication, and load-based traffic redirection. This plan marks a pivotal evolution in SDN architecture, allowing multiple applications to leverage a unified network infrastructure concurrently, unlike traditional networks requiring device reconfiguration for each application change. Ease of development and integration in the application plane

can introduce vulnerabilities with potential network-wide impact.

Below are the assaults targeting the SDN application plane:

- **Unauthorised access :** Unrestricted application access poses a security risk. Without a robust trust mechanism between applications and the control layer, the SDN controller can't distinguish legitimate applications from regular network services. This vulnerability creates an opportunity for attackers to bypass security measures and inject malicious code. This results in the controller treating all applications uniformly. Such unauthorized access to diverse applications can provide attackers with insights into the network's operations, thereby increasing the likelihood of exploiting different network components [19].
- **Disclosure of information through the application server :** After infiltrating the application server, an attacker can retrieve and expose the data of any application that has been active in the RAM, whether presently or in the past. In conventional networks, this type of breach is referred to as a RAM scraper attack. Such attacks target to scan the processes in RAM within the application server to obtain application data via the northbound API. This method enables attackers to discern the controller's rules governing different network flows [19].
- **User privileges modification :** Network virtualization allows each user to customize the network according to their requirements while maintaining isolation. Users are granted specific privileges to run different applications as needed. However, attacker can breach the application server in order to modify these privileges, leading to dysfunctional outcomes [19].

3.4 Attacks on the SDN interfaces

The northbound and southbound interfaces are utilized in centralized controller environments, whereas the westbound and eastbound interfaces are employed in distributed controller environments. These interfaces facilitate the transmission and reception of network information, making them susceptible targets for attackers aiming to eavesdrop on the data.

- **Attacks on Southbound Interface :** In SDN, the southbound interface often relies on the OpenFlow (OF) protocol. This allows OF switches to communicate with the controller. Whenever a switch encounters a new packet, it sends a *Packet_In message* to the controller via the southbound interface. This message exchange makes the southbound interface a valuable tool for extracting information about network traffic. Security flaws in TLS encryption could grant attackers a foothold in the southbound interface, potentially compromising network control, enabling them to manipulate flow rules by creating, modifying, or deleting them. This manipulation can lead to network malfunctions due to the introduction of malicious flow entries into the flow table. Additionally, attackers can flood OF switches with forged packets, each bearing a unique identity, prompting the switches to generate an excessive number of *Packet_In messages*. The high volume of messages can saturate the communication channel between the OF switches and the controller, hindering communication and potentially leading to performance degradation.
- **Attacks on Northbound Interface :** The northbound interface facilitates communication between the applications of the application plane and the controller. The northbound interface, unlike the southbound one with its established OF protocol, remains unstandardized due to its relative infancy. This lack of uniformity creates potential vulnerabilities that attackers could exploit

to disrupt communication between applications and the controller. Unauthorized access to the northbound interface could allow attackers to delete crucial information, resulting in the generation of falsified outputs by the application.

- Attacks on eastbound and westbound Interfaces : The eastbound and westbound interfaces are susceptible to a range of attacks. Attackers with unauthorized access to the management console can exploit the information updates transmitted through these interfaces. They may take advantage of unencrypted data communication between controllers for sharing network information updates. Unencrypted communication between controllers through the northbound interface presents another attack vector. Interception of this data could allow attackers to steal sensitive information or manipulate communication.

3.5 Miscellaneous Threats

A successful topology poisoning attack creates a domino effect, leaving the network vulnerable to a cascade of malicious activities. These include hijacking communications (MITM attacks), overwhelming the network (DoS attacks), impersonation (identity spoofing), and even denial of responsibility (repudiation). Table 3.1 explore how various threats can impact topology discovery in relation to the CIA triad (Confidentiality, Integrity, and Availability).

3.6 Topology Discovery Threats

Within a topology poisoning attack, a specific tactic leverages a compromised system to impersonate a legitimate host's location on the network. This "host location hijacking" allows the attacker to intercept the target host's traffic. Exploiting the

Table 3.1: SDN Topology discovery threat attacks

Threats/ Attacks	Reasons	Affects	CIA		
			C	I	A
MITM	Fake link injection	Modifies the shortest path	Yes	Yes	No
DOS	Removal of legitimate link	Workload increases	No	No	Yes
Identity Spoofing	Spoofing host identity	Unauthorised information access	Yes	Yes	No
Repudiation	LLDP packet Spoofing	Obfuscating attacker origin	Yes	Yes	No

lack of authentication in the Host Tracking System (HTS) used by the controller for host mobility management in SDN, attacker manipulates the network by spoofing the target host's location data, rerouting traffic intended for the legitimate host to their own machine. This manipulation disrupts topology-dependent applications like routing and load balancing. Table 3.1 shows the classifications of the topology poisoning attacks:

- Attack entity
 - *Host-based attacks*: In a host location hijacking attack, a compromised device on the network impersonates the target host's location. This allows the attacker to intercept traffic meant for the legitimate host. By exploiting the Controller's Host Tracking System (HTS), which lacks authentication for host mobility in SDN, the attacker can alter the host's location information. These vulnerabilities can manipulate the SDN controller. Malicious actors could inject spoofed packets to create inaccurate switch profiles, disrupting applications that rely on network topology information like routing and load balancing. Additionally, attackers might forge legitimate Link Layer Discovery Protocol (LLDP) packets, potentially creating phantom network connections between switches. This ma-

nipulation deceives the controller, potentially causing it to route traffic through fabricated links, ultimately delivering it to the attacker's machine.

- *Switch-based attacks*: Malicious actors can exploit SDN vulnerabilities by deploying compromised OF switches. These switches forge LLDP packets, creating deceptive network connections. This Link Fabrication Attack (LFA) can have a significant impact, as the compromised switches establish fake links with numerous devices throughout the network. Detecting topology poisoning caused by malicious switches is challenging because there are limited indicators of fake link creation, and no host involvement is required to create these fake links.

Malicious OF switches can disrupt network topology by manipulating LLDP packets. Instead of forwarding LLDP packets from one switch to the controller (as intended), the attacker redirects them to another malicious switch. This second switch then relays them as *Packet_In* messages to the controller, creating a false impression of a direct link between the two malicious switches. This deception can trick the controller into making decisions based on a fabricated network layout.

- Controller vulnerabilities

- *Host Tracking System (HTS)*: HTS vulnerabilities allows attackers to manipulate host locations. Unverified host updates via *Packet_In* messages expose a vulnerability. Earlier Floodlight and OpenDaylight controllers lacked proper authentication, allowing attackers to spoof host identities through a permissive API. The controller, assuming these updates as genuine, adjusts the host profiles accordingly. This lack of authentication disrupts services, particularly routing.

- *SDN Link Discovery Vulnerability:* SDN relies on LLDP for switch communication but lacks authentication and path verification, leaving it vulnerable to fake link attacks or LFA.

3.7 Conclusion

In conclusion, the implementation of SDN in IP multicast systems presents both advantages and challenges in terms of security. While SDN offers centralized control, granular access control, dynamic adaptability, end-to-end encryption, and enhanced visibility, it also introduces new attack vectors across various planes of the SDN architecture.

The various attacks on the SDN planes and interfaces pose significant threats such as malicious OF switches, fraudulent flow rule insertion, link flooding attacks, topology discovery attacks, unauthorized access to applications, and attacks on SDN interfaces. Additionally, miscellaneous threats like MITM attacks, DoS attacks, identity spoofing, and repudiation further exacerbate security concerns.

In essence, while SDN-enabled IP multicast systems offer enhanced security features compared to traditional multicast systems, proactive measures must be taken to address the evolving threat landscape and safeguard against potential vulnerabilities, ensuring the integrity, confidentiality, and availability of network resources and data transmission.

4 Limitations in current SDN Solutions

While advanced SDN solutions have made significant strides in addressing the limitations of traditional IP multicast, several challenges remain. The advanced SDN solutions aim to enhance security, scalability, and efficiency in broadcasting, however, despite its advancements, certain limitations persist.

4.1 Security Limitations

One of the key limitations lies in the security measures implemented within these advanced SDN solutions. While features like encryption, Access Control Lists (ACLs), Role-Based Access Control (RBAC), and integrated firewall capabilities bolster security, they may not offer comprehensive protection against evolving threats. Advanced SDN solutions may lack advanced analytics for traffic analysis and anomaly detection, leaving networks vulnerable to sophisticated attacks. Additionally, the absence of alerting mechanisms and customizable reporting for network troubleshooting can hinder timely response to security incidents.

4.2 Scalability Challenges

Another limitation relates to the scalability of these advanced SDN solutions, particularly in large networks with numerous devices. While Arista's solutions excel in performance and bandwidth management, they may face scalability challenges when it comes to monitoring capabilities. For instance, Arista CloudVision Portal (CVP) prioritizes automation and orchestration over extensive monitoring features. This may result in limitations in efficiently scaling monitoring capabilities to meet the demands of expansive networks.

4.3 Monitoring and Analytics

Furthermore, the monitoring and analytics capabilities of current solutions may not fully meet the requirements of dynamic multicast environments. For example, Arista's MCS provides real-time monitoring and telemetry, it may lack advanced analytics for in-depth traffic analysis and anomaly detection. This shortfall can impede the identification of emerging threats and the timely resolution of network issues, ultimately impacting network performance and reliability.

4.4 Automation and Orchestration Focus

Moreover, there is a notable focus on automation and orchestration in current solutions, potentially at the expense of comprehensive monitoring and analytics. While automation streamlines network provisioning and management tasks, it may overshadow the need for robust monitoring and analytics capabilities. As a result, networks may lack the agility and responsiveness required to adapt to evolving threats and dynamic traffic patterns effectively.

4.5 Conclusion

In conclusion, while advanced SDN solutions offer significant advancements in addressing the limitations of traditional IP multicast, several challenges persist. These include security limitations, scalability challenges, and shortcomings in monitoring and analytics capabilities. Addressing these limitations will require a concerted effort to enhance security measures, improve scalability, and bolster monitoring and analytics capabilities to ensure the resilience and efficiency of multicast networks in today's dynamic environments.

5 Recommendations for Mitigation Strategies

Building upon the analysis of SDN architecture, vulnerabilities, threats, and attacks presented earlier, this section identifies the security challenges within SDN environments. To address them, following are the novel solutions and associated strategies proposed.

5.1 Network segmentation

Problem: One of the most commonly MITM attack involves injecting a counterfeit link and deceiving the SDN controller into redirecting traffic through the attacker's location. This compromises the integrity and confidentiality of data transmitted through the bogus link, allowing for unauthorized access to the controller in an SDN environment. Subsequently, the attacker gains control over network devices and traffic flow, enabling interception and modification of communication between parties, thus tampering with the transmitted data.

Solution: To mitigate such attacks, the implementation of a FlowVisor approach is recommended. FlowVisor serves as a virtualization layer positioned between the SDN controller and the underlying network infrastructure. It divides network resources into distinct virtual slices, each managed by an independent controller instance. This setup enables the creation of isolated network segments, granting dif-

ferent tenants or applications their own virtualized network view and control over allocated resources without affecting others. FlowVisor facilitates multi-tenancy, resource isolation, and heightened network security within SDN deployments.

5.2 Detection and response mechanism using devices

Problem: Addressing the susceptibility of SDN environments to MITM attacks, these attacks exploit the vulnerability of spoofed Link Layer Discovery Protocol (LLDP) packets, typically generated by hosts rather than switches. These fake packets can deceive the network into redirecting traffic through the attacker's location, compromising the confidentiality and integrity of data transmission.

Solution: A set of devices (switches or hosts) can be leveraged to identify such suspicious LLDP packets. Legitimate LLDP communication typically occurs only between switches, so any LLDP packets generated by hosts will be flagged as suspicious. Additionally, analyzing regular network traffic (TCP/UDP) can help distinguish between device types. If a device is identified as a host, any topology updates originating from it should be disregarded as illegitimate. This proactive detection and response mechanism effectively halts the MITM attack in its early stages by identifying and isolating the malicious host within the SDN.

5.3 Monitoring network connectivity by updating flow graph

Problem: A critical vulnerability within SDNs stems from inadequate authentication mechanisms for LLDP packets, paving the way for repudiation attacks. Malicious actors exploit this flaw by forging and injecting fake links, deceiving the

controller into recognizing fictitious connections. Subsequently, they can disavow any involvement, causing confusion within the affected network. Moreover, the alteration of LLDP packet values compromises both data confidentiality and integrity. The consequences are severe, with injected fake links disrupting network traffic flow and potentially instigating DoS attacks.

Solution: The recommended solution to mitigate such attacks entails developing a defense strategy that involves creating an updated flow graph using metadata from Packet_In and FEATURES_REPLY (OF protocol messages) to monitor network connectivity and identify abnormalities. Furthermore, implementing MAC-IP address binding through a policy engine ensures consistency, thereby facilitating the detection of deviations caused by spoofed LLDP packets.

5.4 Network protection against LLDP-based attacks

Problem: Due to the vulnerabilities in link discovery, several attack methods pose risks within SDN environments. Firstly, attackers may engage in LLDP packet modification by falsifying switch IDs, thereby creating deceptive links between switches to deceive the controller. Secondly, attackers can employ LLDP packet relay, forwarding legitimate LLDP packets to other switches to fabricate false link impressions. The consequences of such attacks are significant, potentially resulting in misconfigured topology within the controller, leading to DoS attacks, MITM attacks, and routing issues.

Solution: Following are the recommended mitigation strategies aimed at reinforcing defenses against LLDP-based attacks :

- *LLDP Packet Authentication:* Implementing LLDP packet authentication enables the network to verify the origin and integrity of LLDP packets, ensuring that only authorized devices can participate in network communications.

- *Hybrid Trunk Port Security (HTS)*: Enabling HTS on internal switch ports adds an extra layer of security by restricting communication to authorized devices only. This helps prevent unauthorized access or communication from devices that may attempt to exploit vulnerabilities in the network.
- *Securing Tunnel Endpoints*: It's crucial to secure tunnel endpoints to safeguard multi-hop connections, particularly in environments where SDN or tunneling protocols are used extensively. Robust authentication mechanisms should be employed to ensure that only authorized entities can establish connections and exchange data over these tunnels.
- *Network Activity Monitoring*: Continuous monitoring of network activity is essential for detecting and responding to any suspicious LLDP traffic promptly. This involves analyzing network logs, traffic patterns, and anomalies to identify potential security threats or unauthorized activities in real-time, allowing for immediate action to mitigate risks and protect the network infrastructure.

6 Proposed Solution: Real-time event monitoring policies framework

The ever-increasing popularity of high-bandwidth network traffic services like live streaming and video-on-demand is creating a surge in data for broadcast companies using IP multicast for content delivery. This, combined with the expanding capabilities of SDN, results in a vast amount of network data. But this data isn't simply a byproduct – it's a powerful tool. By analyzing network data, SDN-based IP multicast system administrators gain valuable insights into the health and performance of their infrastructure. This data acts as a window, allowing them to assess the Quality of Service (QoS) for multicast streams. They can understand how traffic flows across the network in real-time, identifying bottlenecks and potential issues that could disrupt smooth content delivery.

This information is crucial for proactive network management. Network administrators can leverage it to perform several critical tasks. Firstly, they can quickly pinpoint network faults that could disrupt multicast streams, ensuring a smooth viewing experience for audiences. Secondly, by analyzing network performance data, administrators can identify the most efficient paths for multicast streams, minimizing latency and jitter through optimized routing strategies. Finally, network data can reveal suspicious activity that might target the multicast system. This allows administrators to take action and protect content distribution by mitigating cyber-

attacks.

However, collecting all raw data from every network device can overwhelm processing and analysis systems. While compression and sampling techniques can alleviate this problem by reducing data volume, they might compromise the accuracy of further analysis. Therefore, developing efficient monitoring methods that can collect this huge data and analyse it, specifically designed for SDN-based IP multicast systems is essential. These methods should aim to gather the minimal amount of real-time network traffic data necessary, while still ensuring the quality of data for further analysis. Striking this balance is critical for unlocking the full potential of network data and ensuring the smooth and reliable delivery of content through IP multicast systems.

6.1 Related work

This section provides a brief overview of the current state of the art in monitoring of network traffic within SDN architecture comprising data collection, data analysis, visualization and administration and management of huge data.

Monitoring in a system is one of the initial stages, functionally delineating the capabilities of an intrusion detection and prevention system. Hence, numerous associated subjects and scientific contributions have undergone investigation. Maintaining continuous surveillance of network behaviour aids in establishing and executing more dependable system configurations [21]. In one of the researches, a consolidated cloud monitoring classification is suggested among various other considerations [22]. In another research, the characteristics of a monitoring system in comparison to intricate contemporary cloud infrastructures is examined [23] while some of the solutions are designed in Kubernetes [24]. Moreover, pertinent literature delineates architectures tailored to distinct attack patterns, exemplified in an approach that provides a real time monitoring solution for malicious attacks [25].

Libpcap, a cornerstone open-source library, provides a user-level interface for capturing network packets on various systems. This functionality empowers numerous network monitoring tools, including Wireshark and Tcpdump, to analyze and understand network traffic. Despite its utility, it cannot address certain open issues in network data collection. NetFlow, another classical method, collects packets on switches but struggles with resource constraints, particularly during high traffic or DDoS attacks. Despite mitigation methods, NetFlow can still overwhelm switches, even when sampling is used. Sketch-based data collection offers a more efficient alternative by counting packets with hash tables. OpenSketch [26] and other approaches [27], [28], [29] demonstrate that sketches can efficiently detect DDoS attacks from elephant flows. However, these methods have limitations, including ineffectiveness against certain attacks, deployment at fixed nodes like gateways without sensing global traffic, and potential false positives from hash collisions. Jing et al. [30], [31] introduced reversible sketch-based methods to address some of these issues but lack adaptability and automatic data collection strategies. Flowsense passively detects flow removal messages in switches but may lead to overload, and methods based on active measurement or adaptive statistics collection like PayLess can cause data redundancies due to coarse-grained filtering or high data volume. Each of these methods has specific strengths but faces unique challenges in adaptability, resource efficiency, and vulnerability to various network threats. D. Zhou et al. [2019], introduces an adaptive system that dynamically selects optimal data collection nodes based on network status. Additionally, it performs flow sampling considering flow characteristics to reduce data volume while maintaining accuracy. While effective against mainstream DDoS attacks, the system might not detect other network attack types [32].

Existing traffic monitoring approaches in Software-Defined Networking (SDN) often struggle to balance probing frequency (how often data is collected) and monitoring

accuracy. This imbalance can lead to excessive resource consumption or inaccurate data. To address this, various studies have proposed different methods and frameworks to optimize SDN traffic monitoring and related network operations.

For instance, E. F. Castillo et al. [33] introduced IPro, a system that utilizes Knowledge-Defined Networking and Reinforcement Learning to optimize probing frequency in SDN traffic monitoring. The approach aims to balance minimizing control channel overhead and controller CPU usage while maintaining high monitoring accuracy. However, one notable drawback is its lengthy convergence time of approximately 238 seconds. Similarly, A. Khalid et al. [34] proposed an SDN-based system for live video streaming that leverages SDN's dynamic control to optimize resource usage for Internet Service Providers (ISPs) and Content Delivery Networks (CDNs). This system employs a cross-layer approach that combines network-assisted multicast with dynamic bitrate adaptation, showing significant improvements in user goodput and reduced frame drops. Nevertheless, the authors do not address potential limitations, such as implementation complexity or scalability for large user bases. In another approach, R. Mohammadi et al. [35] addressed Quality of Service (QoS) challenges in SDN-based multicast applications by proposing an adaptive traffic engineering method. This method uses an Integer Linear Programming (ILP) problem and an ant colony algorithm to find optimal resource allocation. Although this solution outperforms traditional protocols, the paper does not explore computational complexity or scalability concerns for large networks. Meanwhile, B. Wang et al. [36] introduced FlexMonitor, a flexible monitoring framework for SDN, which offers diverse deployment methods and an increased variety of data sources. Despite these advantages, the authors acknowledge limitations in interpreting monitoring tasks from applications, with future plans to improve these interpretations. Another significant contribution is from S. A. R. Shah et al. [37], who proposed a solution for monitoring individual controller load in centralized SDN architectures

to prevent under-utilization or overloading. This solution combines an ONOS-based collection manager with a collectd-based agent on controllers to gather real-time statistics. However, the paper acknowledges that it only addresses monitoring and plans to extend the work to include dynamic load balancing and scheduling for efficient workload distribution within the controller cluster. Building on this concept, Zaalouk et al. [2014] proposed OrchSec, an architecture that leverages an orchestrator to enhance network security by utilizing SDN control functions [38]. This architecture separates control and monitoring functions, reducing overhead on SDN controllers. While OrchSec demonstrates its ability to mitigate DNS amplification attacks, further exploration is needed to expand its capabilities for other security applications. In the realm of cloud computing, C. T. Yang et al. [30] proposed an SDN/NFV-based traffic monitoring system for Infrastructure as a Service (IaaS) clouds. This system leverages software solutions to achieve cost-efficient deployment and flexible management. Despite matching traditional hardware performance, the authors acknowledge limitations in scalability and plan to test the system in broader network environments with larger cloud workloads. Turning to latency measurement, D. Sinha et al. [39] proposed two methods for measuring latency in SDN: an improved looping technique using IP Time-to-Live and a queue-length technique requiring switch-level implementation. While both methods offer real-time latency reporting, the looping technique needs refinement, and the queue-length technique requires fine-tuning for accuracy.

Similarly, Y.Y. Yang et al. [40] proposed a virtualized traffic monitoring system using SDN and Network Functions Virtualization (NFV), offering cost efficiency by replacing a physical switch with a software-defined one. This approach eliminates the need for additional hardware, yet the authors acknowledge limitations in performance and plan to address these in future work.

Finally, R. Kumar et al. [41] proposed an SDN-based framework to manage real-time

network traffic, aiming to guarantee end-to-end delays through a heuristic algorithm that considers bandwidth and timing requirements. Although the framework has potential, it acknowledges trade-offs in using commercial-off-the-shelf components, which could affect critical systems.

Another interesting approach is from H. Zhou et al. [42], who proposed a real-time method for detecting compromised devices in untrusted SDN networks by leveraging backup controllers to audit network update events. This system offers real-time detection but raises questions about scalability for large SDN networks with many devices and the overhead of information collection by backup controllers.

A review of existing research, including several surveys [43]–[42], highlights the need for a more efficient data collection method in SDN. Ideally, this method would dynamically select a minimal set of nodes for data collection, adapting to network conditions. Hence, there should be an approach based on network status to minimize data collection volume while still enabling high-accuracy detection of sensitive network events.

This thesis presents the design of the framework of a real-time event monitoring policies for network traffic specifically designed to detect and mitigate security threats targeting IP Multicast traffic in SDN environments.

6.2 Framework: SDN-Multicast Security Monitor Policies

The SMSMP framework tackles real-time event monitoring specifically for securing IP multicast traffic within SDN environments. It addresses the unique challenges of multicast security by leveraging a powerful combination of advanced techniques:

6.2.1 Advanced Techniques

- **Machine Learning-based Threat Detection:** SMSMP utilizes machine learning algorithms trained on historical network traffic data and threat intelligence feeds. These algorithms can be tailored to identify anomalies in multicast traffic patterns, such as sudden spikes in traffic volume, unexpected source or destination addresses, or deviations from established baselines. Popular algorithms for anomaly detection include One-Class SVMs, Isolation Forests, or LSTM networks.
- **In-depth Flow Analysis:** SMSMP performs in-depth flow analysis specifically targeted towards multicast groups. It examines flow rules and statistics associated with these groups to detect potential anomalies and DoS attacks. This analysis might involve techniques like:
 - Identifying inconsistencies between flow rules and actual traffic patterns.
 - Detecting unusual flow rates or packet sizes associated with specific multicast groups.
 - Monitoring for rapid changes in the number of devices joining or leaving multicast groups.
- **Optional SDN Controller Integration and Automated Response:** For more immediate mitigation, SMSMP can optionally integrate with the SDN controller. Upon threat detection, this integration allows for automated responses to be triggered through the controller's southbound API (e.g., OpenFlow). Examples of automated responses include:
 - **Device Isolation:** Isolate compromised devices suspected of malicious multicast traffic generation.

- Traffic Rerouting: Reroute multicast traffic away from vulnerable network segments or specific devices.
- Flow Rule Updates: Dynamically update flow rules to block malicious traffic or restrict access to specific multicast groups.

6.2.2 Threat Modeling Integration and Continuous Improvement

The SMSMP framework integrates seamlessly with existing threat modeling approaches, as depicted in (Figure 6.1). This integration strengthens the overall security posture by aligning threat detection and mitigation strategies with the identified vulnerabilities and potential attack vectors within the network environment.

To ensure continuous improvement and refinement of the threat modeling framework, several mechanisms can be established:

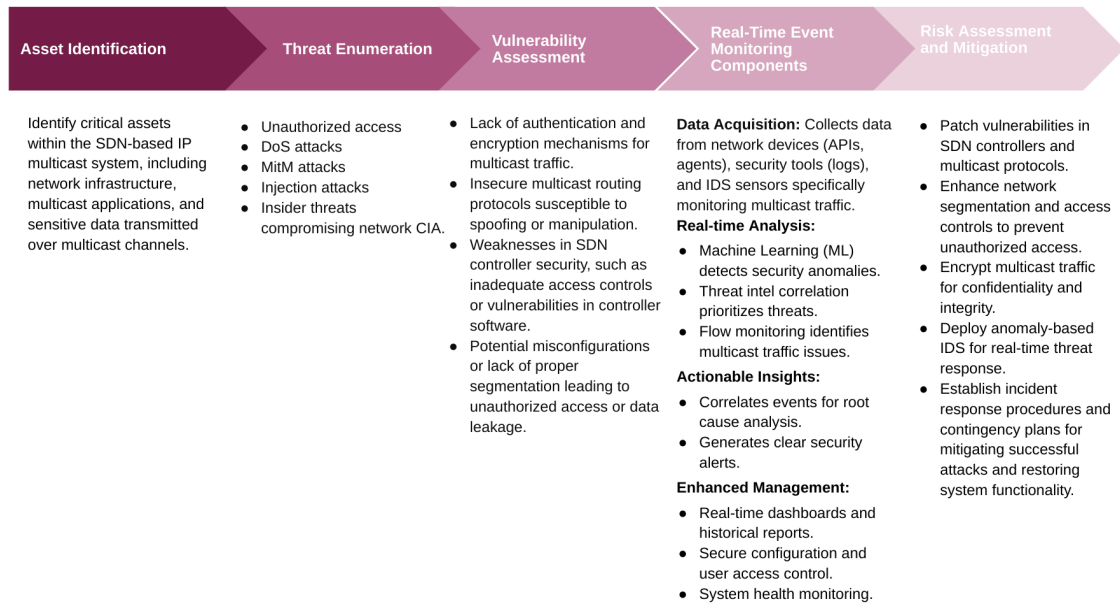
- Regular Threat Profile Review: Regularly review and update threat profiles to reflect the evolving threat landscape and emerging vulnerabilities specific to multicast communication.
- Vulnerability Assessment and Management: Conduct periodic vulnerability assessments of network devices and software components involved in multicast traffic to identify and address potential security weaknesses.
- Mitigation Strategy Adaptation: Adapt mitigation strategies within the SMSMP framework based on newly identified threats and evolving network requirements. This may involve retraining machine learning models or implementing new automated response mechanisms.

6.2.3 ML, Real-time Monitoring, and Threat Intelligence

By combining machine learning with real-time event monitoring and threat intelligence, the SMSMP framework offers a comprehensive and automated approach to securing IP multicast traffic within SDN environments. This combination provides several advantages:

- **Enhanced Anomaly Detection:** Machine learning algorithms can identify complex and evolving threats that might be missed by traditional signature-based detection methods.
- **Faster Response Times:** Real-time event monitoring enables immediate detection and response to security threats, minimizing potential damage.
- **Improved Threat Context:** Integration with threat intelligence feeds provides valuable context about known threats and vulnerabilities, allowing for more targeted mitigation strategies.
- **Regular Threat Profile Review:** Regularly review and update threat profiles to reflect the evolving threat landscape and emerging vulnerabilities specific to multicast communication.
- **Vulnerability Assessment and Management:** Conduct periodic vulnerability assessments of network devices and software components involved in multicast traffic to identify and address potential security weaknesses.
- **Mitigation Strategy Adaptation:** Adapt mitigation strategies within the SMSMP framework based on newly identified threats and evolving network requirements. This may involve retraining machine learning models or implementing new automated response mechanisms.

Figure 6.1: Threat modelling framework for SDN incorporating SMSMP

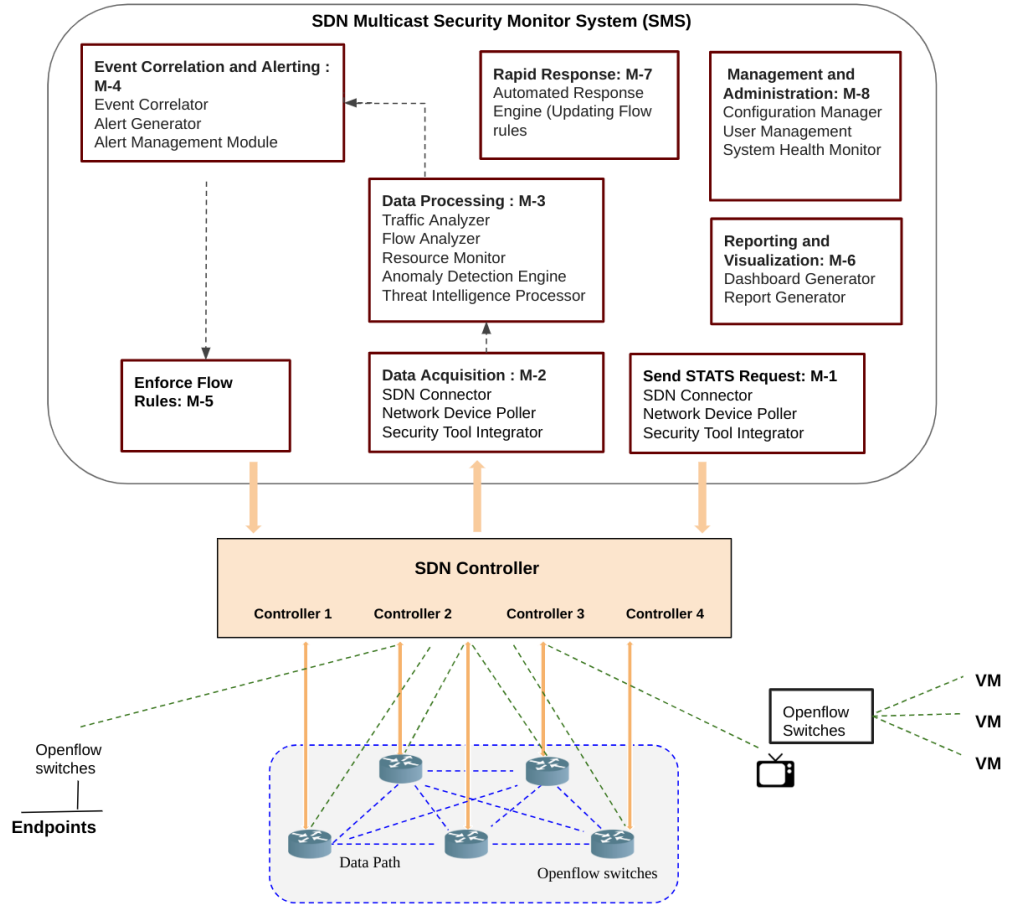


6.2.4 SMSMP Application Model integrated with SDN

The proposed SMSMP framework (Figure 6.2) introduces several distinctive features compared to other proposed models. SMSMP, powered by machine learning, offers real-time anomaly detection with explainability, aiding in understanding and addressing underlying causes effectively. It conducts contextual threat analysis to prioritize threats and deliver actionable insights, and employs predictive analytics for proactive threat prevention by analyzing historical data and identifying patterns. With a decentralized and scalable architecture, SMSMP reduces latency and enhances scalability, while integrating with network automation tools for real-time response to security threats, such as rerouting traffic or isolating infected devices upon detecting a DoS attack. Users can define their own threat models for tailored monitoring and targeted alerting. Leveraging advanced techniques like statistical flow analysis or traffic fingerprinting, SMSMP can analyze encrypted traffic without decryption, addressing a common monitoring challenge. It can also employ federated learning for decentralized threat intelligence, enabling improved threat

detection accuracy while maintaining data privacy. Furthermore, SMSMP enables real-time threat hunting with interactive visualization, integrates bio-inspired adaptive response mechanisms for dynamic threat response, and utilizes blockchain-based secure event logging for immutable security event records. Additionally, adopting a zero-trust approach with continuous micro-segmentation enhances network security by continuously monitoring and segmenting the network based on real-time security assessments, limiting the lateral movement of attackers.

Figure 6.2: SMSMP integrated with SDN



Primarily, the SMSMP System continuously monitors network activity, identifying and flagging suspicious events such as unauthorized access attempts or DoS attacks. Integrated with the SDN controller, it enables real-time threat detection and re-

sponse, enhancing network security and mitigating potential risks promptly (Figure 6.2). It consists of 8 modules and each module performs the following operations:

- **Data Acquisition (M-1, M-2)**

The data acquisition phase involves two key components, each with distinct functions. *The first component, M-1: Data Collection Agent*, continuously gathers current network status information from network devices at regular intervals. This data includes network traffic metrics, device status, and other relevant information to maintain an up-to-date view of network operations. *The second component, M-2: Central Data Hub*, serves as a central repository for data collected from various sources. It integrates information from SDN control plane APIs, providing real-time network state, and may also include data from optional network device pollers, which offer deeper device-specific insights. Additionally, the Central Data Hub can incorporate data from security tools, either through direct API integration or by parsing log files, further enriching the data set for subsequent analysis.

- **Security Analysis (M-3)**

The security analysis phase (M-3) focuses on identifying potential security threats through the analysis of the collected data. This phase employs various techniques to detect anomalies or malicious behavior. Traffic analysis is used to scrutinize network patterns for unusual activity, while flow monitoring helps track specific data flows to detect deviations from expected behavior. Machine learning techniques can be applied to identify patterns indicative of security threats, and integration with threat intelligence sources provides additional context for recognizing known threats.

- **Event Correlation and Alerting (M-4)**

After the security analysis phase, the event correlation and alerting phase (M-

4) uses the processed data from the Central Data Hub to identify and respond to potential security incidents. Event correlation involves cross-referencing data to uncover security incidents and determine their root causes, facilitated by an Event Correlator. Once an incident is identified, the Alert Generator creates alerts based on the severity and type of threat, and the Alert Management Module prioritizes and routes these alerts for efficient response. This phase ensures that potential threats are not only detected but also acted upon promptly and effectively.

- **Threat Mitigation (M-5,M-7)**

The threat mitigation phase, incorporating components M-4 and M-5, is designed to take swift action against detected security threats. Upon identifying malicious traffic, M-4 enforces flow rules via M-5 to either block or reroute the traffic, thereby reducing the risk of further propagation. The Automated Response Engine (M-7) facilitates this process by initiating predefined actions, such as updating flow rules in the SDN controller to isolate infected devices or redirect traffic away from compromised sections of the network. These automated responses ensure a rapid reaction to potential threats, minimizing response time and mitigating the impact of security incidents.

Beyond these core components, the monitoring system also includes Rapid Response Modules (M-7). The Automated Response Engine, part of this module, allows for predefined responses to detected threats, including updating flow rules to manage network traffic without depending on real-time data flow. This flexibility ensures robust mitigation measures are in place.

The system gathers network-related data, including flow tables and switch configurations, from the SDN control plane via APIs. It also retrieves information from lightweight monitoring agents installed on network devices, providing additional context and deeper insights into network operations. Furthermore, the

system integrates seamlessly with various security tools like firewalls and IDS/IPS, extracting valuable data through APIs or log parsing techniques. This comprehensive data collection approach ensures the system remains robust in identifying and mitigating security threats.

- **Reporting and Visualization (M-6)**

The reporting and visualization phase (M-6) utilizes the processed data to create user-friendly visual representations of network activity and security events. Real-time dashboards, generated by the Dashboard Generator, offer a comprehensive view of network status and ongoing security events. Historical reports, created by the Report Generator, provide documentation for audits, compliance, and trend analysis. This phase is crucial for both operational oversight and compliance with industry standards.

- **Management and Administration (M-8)**

The management and administration phase (M-8) includes functions that, while separate from real-time data flow, are critical for system management and maintenance. The Configuration Manager allows administrators to configure data sources, monitoring rules, and user access control. To safeguard sensitive data, the User Management component enforces role-based access control. This ensures only authorized personnel with the appropriate permissions can access this information. Finally, the System Health Monitor oversees the health and performance of the SMSMP application, ensuring it operates reliably and efficiently.

6.2.5 Key Advantages of SMSMP

The proposed SMSMP framework offers a robust approach to real-time traffic monitoring with several key advantages over existing solutions.

- Machine Learning-powered Anomaly Detection with Explainability: Offers real-time anomaly detection with insights into underlying causes for effective mitigation.
- Contextual Threat Analysis and Actionable Insights: Prioritizes threats and delivers actionable intelligence for informed decision-making.
- Predictive Analytics for Proactive Threat Prevention: Analyzes historical data to identify patterns and prevent future threats.
- Decentralized and Scalable Architecture: Reduces latency and enhances scalability for large networks.
- Network Automation Integration for Real-Time Response: Integrates with network automation tools for immediate response to security threats.
- Customizable Threat Models and Monitoring: Allows user-defined threat models for tailored monitoring and targeted alerting.
- Encrypted Traffic Analysis without Decryption: Analyzes encrypted traffic using advanced techniques like statistical flow analysis or traffic fingerprinting.
- Federated Learning for Decentralized Threat Intelligence: Improves threat detection accuracy while maintaining data privacy.
- Real-Time Threat Hunting and Bio-inspired Adaptive Response: Enables interactive visualization for threat hunting and utilizes bio-inspired mechanisms for dynamic threat response.
- Blockchain-based Secure Event Logging: Provides immutable security event records for enhanced security.

- Zero-Trust Approach with Continuous Micro-segmentation: Continuously monitors and segments the network based on real-time assessments, limiting attacker movement.

Overall, SMSMP offers a comprehensive and adaptable solution for real-time traffic monitoring and security in SDN networks.

6.2.6 Implementation of SMSMP

Implementing the monitoring system in an SDN environment involves the following key steps:

Step 1: Controller Integration (Northbound API)

SDN Controller Compatibility: Ensure compatibility with the target SDN controller (e.g., OpenFlow, ODL) by leveraging its northbound APIs. Popular controllers offer well-documented APIs for programmatic access.

Data Acquisition Libraries: Utilize libraries specific to the chosen controller to interact with its APIs. For example, use the `ryu.app.ofctl` library for interacting with OpenFlow controllers.

Data Streams: Establish real-time data streams for:

- **Network Topology:** Retrieve network device information and connections using controller APIs (e.g., `OFPT_GET_SWITCH_DESCRIPTION`).
- **Flow Statistics:** Continuously collect flow statistics (e.g., packet count, byte count) for traffic analysis using controller APIs. (e.g., `OFPT_FLOW_STATS_REQUEST`).
- **Control Plane Updates:** Monitor controller logs or utilize API notifications to capture control plane events (e.g., flow rule changes, switch configuration updates).

Step 2: Packet Capture and Analysis (Southbound Path)

Monitoring Probes or Agents: Deploy lightweight agents on strategic network devices (e.g., switches, routers) to capture packets. These agents can leverage libraries like libpcap for packet capture.

Multicast Traffic Identification: Analyze captured packets to identify multicast traffic based on destination MAC addresses.

Deep Packet Inspection (Optional): For deeper analysis, consider deploying probes with Deep Packet Inspection (DPI) capabilities to inspect application layer protocols within multicast traffic.

Step 3: Anomaly Detection and Threat Intelligence Integration

Machine Learning Techniques: Implement anomaly detection algorithms like One-Class SVMs or Isolation Forests, trained on historical network traffic data, to identify deviations from normal multicast behavior.

Threat Intelligence Feeds: Integrate with threat intelligence feeds (e.g., STIX/-TAXII) to access known malicious IP addresses and attack signatures for threat correlation.

Anomaly Scoring System: Develop a scoring system to assign severity levels to anomalies based on factors like frequency, deviation from baseline, and correlation with threat intelligence.

Step 4: Automated Response Mechanisms (SDN Southbound API)

SDN Flow Rule Manipulation: Upon detecting threats, utilize the SDN controller's southbound API (e.g., OpenFlow) to dynamically update flow rules:

- **Traffic Rerouting:** Redirect suspicious multicast traffic away from vulnerable network segments.
- **Traffic Blocking:** Block malicious multicast traffic identified through threat intelligence or exceeding anomaly thresholds.

Security Countermeasure Deployment: Integrate with security tools (firewalls, ID-

S/IPS) to trigger actions like source IP blocking or application layer filtering based on threat severity.

Step 5: Visualization and Reporting

Interactive Dashboards: Develop real-time dashboards using libraries like Grafana or Kibana to visualize network traffic patterns, anomaly heatmaps, and threat alerts.

Detailed Reports: Generate historical reports with insights on network activity, detected anomalies, and security incidents for auditing, compliance purposes, and security team awareness.

Step 6: Continuous Monitoring and Optimization

Real-time Monitoring Agent: The SMSMP system continuously monitors network activity through controller APIs and deployed agents.

Adaptive Anomaly Detection: Periodically retrain anomaly detection models with fresh network traffic data to adapt to evolving network patterns and potential new threats.

Performance Optimization: Regularly monitor the performance of the SMSMP application and agents to identify bottlenecks and optimize resource utilization.

Step 7: Integration with SDN Security Policies

Security Policy Alignment: Ensure the SMSMP application's threat detection and response mechanisms align with existing SDN security policies and protocols.

Policy Enforcement Automation: Automate the enforcement of security policies through integration with the SDN controller's policy management framework.

Continuous Security Evaluation: Conduct periodic reviews of the overall security posture and adapt policies and SMSMP configurations as needed to maintain a robust security posture.

Additional Considerations:

Scalability: Choose technologies and architectures that can scale efficiently to handle large networks with high traffic volumes.

Security: Implement robust security measures for the SMSMP application itself, including secure communication channels, access controls, and vulnerability management practices.

By following these steps and considering the additional aspects, a robust and effective SMSMP application can be implemented for real-time traffic monitoring and security within the SDN environment.

6.2.7 Tools for implementing

To translate the SMSMP framework’s design into a practical system, various tools have been meticulously chosen and integrated. Table 6.1 showcases these tools, demonstrating the broad spectrum of technologies that underpin the framework’s real-time monitoring abilities. By leveraging a combination of open-source and potentially commercial solutions, the framework addresses critical functionalities like data acquisition, analysis, threat detection, and response.

Table 6.1: Description of Modules and Tools

Module	Tools
Data Acquisition	Network Devices: OpenFlow, REST APIs, SNMP
	Security Tools: SIEM
	IDS Sensors: Snort
Data Processing and Threat Detection	Machine Learning Engine: TensorFlow, PyTorch
	Threat Intelligence Integration: Threat intelligence tools (Cisco Talos, Palo Alto)
	Flow Monitoring Module: OpenFlow switch vendors' APIs
Event Correlation and Alerting	Event Correlation Engine: SOAR
	Alerting System: SIEM and SOAR alert systems
Reporting and Visualization	Real-time Dashboard: Grafana, Kibana

6.2.8 Success Factors Metrics

A critical factor in the framework's effectiveness is achieving its defined goals. Table 6.2 elaborates various metrics that can be used to evaluate the success of the SMSMP framework in securing IP Multicast traffic within SDN environments. These metrics will provide valuable insights into the system's performance, efficiency, and overall impact on security posture.

Table 6.2: Success Factor Matrix for SMSMP Security Framework

Success Factor	Metrics	Technical Content
Threat Detection and Prevention	Number of Security Incidents Detected	Consider subcategories for specific threats like DoS attacks, unauthorized source participation, or content injection attempts.
	Time to Detection (TTD)	Integrate with real-time monitoring tools and utilize efficient anomaly detection algorithms.
	False Positive Rate	Employ machine learning techniques with high accuracy and regularly review/refine models.
	Number of Prevented Security Incidents	Monitor effectiveness of automated responses in mitigating threats.
System Performance and Efficiency	Mean Time Between Failures (MTBF)	Implement high availability mechanisms and conduct regular system health checks.
	Resource Utilization	Optimize SMSMP application and consider distributed processing/resource scaling.
	Alert Processing Time	Utilize efficient data processing techniques and optimize communication channels.
Additional Considerations	Security Incident Detection Rate (SIDR)	Complements TTD by calculating the percentage of incidents detected within a specific timeframe.
	Mean Time to Resolution (MTTR)	Tracks the average time to resolve a detected security incident. Analyze trends to improve response procedures.

By monitoring and analyzing these metrics, network administrators can gain valu-

able insights into the effectiveness of the SMSMP framework. This allows for continuous improvement through:

- **Machine Learning Model Refinement:** Continuously incorporating fresh data and integrating the latest threat intelligence feeds can refine machine learning models, improving their detection accuracy and minimizing false positives.
- **Optimization of Automated Responses:** Analyzing the effectiveness of automated responses in mitigating threats allows for fine-tuning rules and improving response strategies.
- **Resource Management:** Monitoring resource utilization helps identify potential bottlenecks and implement resource optimization techniques to ensure smooth system operation.

A data-driven approach to evaluating the SMSMP framework using these metrics empowers network security teams to make informed decisions for maintaining a robust security posture for IP multicast communication within their SDN environments.

7 Conclusion

In conclusion, this thesis has explored the multifaceted landscape of multicast technology and its intersection with SDN, shedding light on both the opportunities and challenges they present, particularly in terms of security. Multicast technology offers a powerful means of data transmission, optimizing network bandwidth and facilitating efficient distribution to multiple recipients simultaneously. However, traditional multicast approaches have faced security vulnerabilities and scalability limitations, necessitating innovative solutions.

The emergence of SDN has provided a promising avenue for addressing these challenges. By decoupling the control, data planes and centralizing control, SDN offers enhanced flexibility, scalability, and security features. Leveraging SDN-enabled IP multicast systems, broadcasters can achieve heightened levels of security, scalability, and flexibility, facilitating seamless live media production workflows and reliable content delivery.

Nevertheless, the adoption of SDN introduces new security considerations and attack vectors across various planes of the SDN architecture. Attacks targeting the SDN infrastructure, control plane, application plane, and interfaces pose significant threats, including malicious switch manipulation, unauthorized access, and DoS attacks. As such, proactive measures must be taken to safeguard against these evolving threats and uphold the CIA (integrity, confidentiality, and availability) triad for network resources and data transmission.

Building upon the analysis of SDN architecture vulnerabilities, threats, and attacks, the proposed framework introduces novel solutions to address security challenges within SDN environments. These solutions include implementing FlowVisor for network segmentation, leveraging switches or hosts for detecting suspicious LLDP packets, developing defense strategies for monitoring network connectivity and updating flow graphs, and employing mitigation strategies against LLDP-based attacks such as packet authentication and continuous network activity monitoring.

Additionally, the SDN-Multicast Security Monitor Policies (SMSMP) framework is proposed, offering real-time event monitoring specifically targeting IP multicast traffic. SMSMP utilizes machine learning, threat intelligence integration, and advanced analytics to provide comprehensive threat detection, proactive prevention, and automated response mechanisms. Its decentralized architecture, scalability, and integration with SDN controllers ensure prompt threat mitigation, enhancing network security in dynamic environments. Evaluation metrics further highlight the framework's effectiveness in securing IP multicast traffic within SDN environments. In essence, while the integration of SDN in IP multicast systems presents significant advantages in terms of security and performance, ongoing vigilance and adaptation to the evolving threat landscape are essential to maintain robust security defenses and ensure the resilience of SDN-enabled networks. Through continuous improvement and proactive security measures, SDN can truly fulfill its potential as a transformative technology for enhancing network security and efficiency in modern broadcast environments.

In future research, I aim to analyze and assess the risk associated with SDN architecture powered by artificial intelligence, particularly focusing on IP multicasting in the live streaming or broadcasting industry, leveraging the knowledge gained during my master's thesis. This research will also entail investigating vulnerabilities within ARISTA EOS Media Control services, including protocols such as RTSP,

RTCP, SDP, and PTP. As these services are pivotal in modern network environments, especially with the increasing integration of multimedia applications and streaming services, a comprehensive assessment will be conducted to uncover potential weaknesses and security gaps that may jeopardize network integrity and data confidentiality. Areas of exploration may involve scrutinizing authentication mechanisms, evaluating encryption methods used in media transmission, and testing for susceptibility to exploitation. By conducting thorough vulnerability assessments and network attack testing, valuable insights into potential attack vectors can be gained, aiding in the development of effective mitigation strategies.

References

- [1] P. Newswire, “Global software-defined networking market to reach \$36.2 billion by 2026.”, 2022.
- [2] S. Islam, N. Muslim, and J. W. Atwood, “A survey on multicasting in software-defined networking”, *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 355–387, 2017.
- [3] H. Subramanian and P. Raj, *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Packt Publishing Ltd, 2019.
- [4] J. Loveless, R. Blair, and A. Durai, *IP Multicast: Cisco IP Multicast Networking*. Cisco press, 2016.
- [5] Cisco, “Software configuration guide, cisco ios xe denali 16.3.x (catalyst 3850 switches) - ip multicast routing technology overview [cisco catalyst 3850 series switches],”, 2018.
- [6] Kahn, *Cisco CCIE Fundamentals: Network Design and Case Studies*. Cisco Press, 1999.
- [7] C. Shannon and D. Moore, “The spread of the witty worm”, *IEEE Security & Privacy*, vol. 2, no. 4, pp. 46–50, 2004.
- [8] T. D. Nadeau and K. Gray, *SDN: Software Defined Networks: An authoritative review of network programmability technologies*. " O'Reilly Media, Inc.", 2013.

- [9] S. Kunić and Z. Šego, “Analysis of television technology transformation from sdi to ip production”, in *2018 International Symposium ELMAR*, IEEE, 2018, pp. 211–216.
- [10] D. G. Dutt, *EVPN in the Data Center*. O’Reilly Media, 2018.
- [11] D. Kreutz, F. M. Ramos, and P. Verissimo, “Towards secure and dependable software-defined networks”, in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 55–60.
- [12] H. S. Abdulkarem and A. Dawod, “Ddos attack detection and mitigation at sdn data plane layer”, in *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, IEEE, 2020, pp. 322–326.
- [13] S. Hameed and H. Ahmed Khan, “Sdn based collaborative scheme for mitigation of ddos attacks”, *Future Internet*, vol. 10, no. 3, p. 23, 2018.
- [14] A. Pradhan and R. Mathew, “Solutions to vulnerabilities and threats in software defined networking (sdn)”, *Procedia Computer Science*, vol. 171, pp. 2581–2589, 2020.
- [15] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, “Multi-controller based software-defined networking: A survey”, *IEEE access*, vol. 6, pp. 15 980–15 996, 2018.
- [16] E. Al-Shaer and S. Al-Haj, “Flowchecker: Configuration analysis and verification of federated openflow infrastructures”, in *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*, 2010, pp. 37–44.
- [17] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, “Modeling and performance evaluation of an openflow architecture”, in *2011 23rd International Teletraffic Congress (ITC)*, IEEE, 2011, pp. 1–7.
- [18] W. Mengmeng, L. Jianwei, C. Jie, M. Jian, and M. Kefei, “Software defined networking: Security model, threats and mechanism”, *Journal of software*, vol. 27, no. 4, pp. 969–992, 2016.

- [19] K. Suleman, “Forensic investigation of link fabrication attack in software defined networks/suleman khan”, Ph.D. dissertation, University of Malaya, 2017.
- [20] K. ElDefrawy and T. Kaczmarek, “Byzantine fault tolerant software-defined networking (sdn) controllers”, in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, IEEE, vol. 2, 2016, pp. 208–213.
- [21] S. Lee, K. Levanti, and H. S. Kim, “Network monitoring: Present and future”, *Computer Networks*, vol. 65, pp. 84–98, 2014.
- [22] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, “Gmone: A complete approach to cloud monitoring”, *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2026–2040, 2013.
- [23] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, “Cloud monitoring: A survey”, *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.
- [24] C.-C. Chang, S.-R. Yang, E.-H. Yeh, P. Lin, and J.-Y. Jeng, “A kubernetes-based monitoring platform for dynamic cloud resource provisioning”, in *GLOBE-COM 2017-2017 IEEE Global Communications Conference*, IEEE, 2017, pp. 1–6.
- [25] D. Kshirsagar and A. Patil, “Blackhole attack detection and prevention by real time monitoring”, in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, IEEE, 2013, pp. 1–5.
- [26] M. Yu, L. Jose, and R. Miao, “Software {defined}{traffic} measurement with {opensketch}”, in *10th USENIX symposium on networked systems design and implementation (NSDI 13)*, 2013, pp. 29–42.
- [27] Q. Huang, X. Jin, P. P. Lee, *et al.*, “Sketchvisor: Robust network measurement for software packet processing”, in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 113–126.

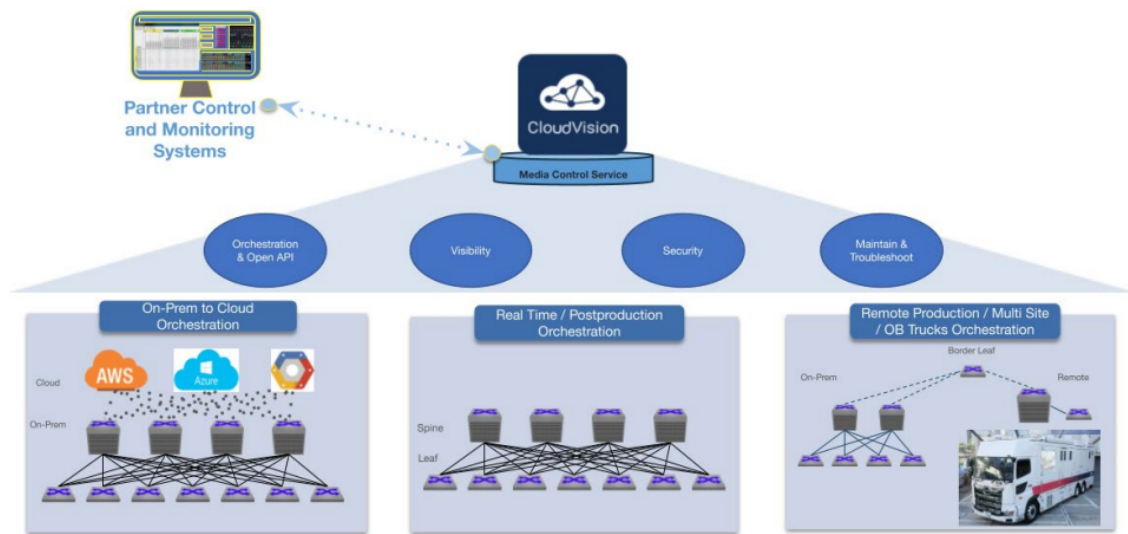
- [28] Q. Huang, P. P. Lee, and Y. Bao, “Sketchlearn: Relieving user burdens in approximate measurement with automated statistical inference”, in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 576–590.
- [29] T. Yang, J. Jiang, P. Liu, *et al.*, “Elastic sketch: Adaptive and fast network-wide measurements”, in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 561–575.
- [30] C.-T. Yang, S.-T. Chen, J.-C. Liu, Y.-Y. Yang, K. Mitra, and R. Ranjan, “Implementation of a real-time network traffic monitoring service with network functions virtualization”, *Future Generation Computer Systems*, vol. 93, pp. 687–701, 2019.
- [31] X. Jing, J. Zhao, Q. Zheng, Z. Yan, and W. Pedrycz, “A reversible sketch-based method for detecting and mitigating amplification attacks”, *Journal of Network and Computer Applications*, vol. 142, pp. 15–24, 2019.
- [32] D. Zhou, Z. Yan, G. Liu, and M. Atiquzzaman, “An adaptive network data collection system in sdn”, *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 562–574, 2019.
- [33] E. F. Castillo, O. M. C. Rendon, A. Ordonez, and L. Z. Granville, “Ipro: An approach for intelligent sdn monitoring”, *Computer Networks*, vol. 170, p. 107 108, 2020.
- [34] A. Khalid, A. H. Zahran, and C. J. Sreenan, “An sdn-based device-aware live video service for inter-domain adaptive bitrate streaming”, in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 121–132.
- [35] R. Mohammadi, R. Javidan, N. Rikhtegar, and M. Keshtgari, “An intelligent multicast traffic engineering method over software defined networks”, *Journal of High Speed Networks*, vol. 26, no. 1, pp. 77–88, 2020.

- [36] B. Wang and J. Su, “Flexmonitor: A flexible monitoring framework in sdn”, *Symmetry*, vol. 10, no. 12, p. 713, 2018.
- [37] S. A. R. Shah, S. Bae, A. Jaikar, and S.-Y. Noh, “An adaptive load monitoring solution for logically centralized sdn controller”, in *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, 2016, pp. 1–6.
- [38] 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 *2014 IEEE Network Operations and Management Symposium (NOMS)*, IEEE, 2014, pp. 1–9.
- [39] D. Sinha, K. Haribabu, and S. Balasubramaniam, “Real-time monitoring of network latency in software defined networks”, in *2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2015, pp. 1–3.
- [40] Y.-Y. Yang, C.-T. Yang, S.-T. Chen, W.-H. Cheng, and F.-C. Jiang, “Implementation of network traffic monitor system with sdn”, in *2015 IEEE 39th annual computer software and applications conference*, IEEE, vol. 3, 2015, pp. 631–634.
- [41] R. Kumar, M. Hasan, S. Padhy, *et al.*, “End-to-end network delay guarantees for real-time systems using sdn”, in *2017 IEEE Real-Time Systems Symposium (RTSS)*, IEEE, 2017, pp. 231–242.
- [42] H. Zhou, C. Wu, C. Yang, *et al.*, “Sdn-rdcd: A real-time and reliable method for detecting compromised sdn devices”, *IEEE/ACM transactions on networking*, vol. 26, no. 5, pp. 2048–2061, 2018.

-
- [43] H. Xie, Z. Yan, Z. Yao, and M. Atiquzzaman, “Data collection for security measurement in wireless sensor networks: A survey”, *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205–2224, 2018.
- [44] ARISTA, *Multicast orchestration with arista media control service*, Available at: <https://www.arista.com/>, n.d.

Appendix A ARISTA Multicast Orchestration

Figure A.1: ARISTA MCS across deployments [44]



Arista's hardware and software provide versatile and scalable features for broadcast and media workflows in various environments. Arista MCS seamlessly integrates into different setups, enabling efficient orchestration of multicast flows across diverse network architectures, including hybrid deployments with PIM/IGMP zones. MCS facilitates real-time/post-production workflows, manages live telecasts, and supports workload migration between cloud and on-premises environments (Figure A.1) [44].

Appendix B MCS API workflow

The broadcast controller connects to MCS's APIs to discover media endpoints and manage network state changes. It configures senders and receivers using MCS's APIs as shown in sample workflows 1:

Listing 1 MCS API sample workflow[44].

```
curl --location --request POST 'http[s]://<mcs server i.p>/mcs/
multicast/programFlows' \--header 'Content-Type: application/json' \
--data-raw '[
  {
    "data": [
      {
        "destinationIP": "225.0.1.2",
        "sourceIP": "35.1.1.4",
        "bandwidth": 25,
        "bwType": "m",
        "inIntfID": "001c.7356.4f12-Ethernet12"
      }
    ],
    "flow-action": "addSenders",
    "transactionID": "GS#1",
    "trackingID": 2098
  },
  {
    "flow-action": "addReceivers",
    "transactionID": "test",
    "trackingID": 2099,
    "data": [
      {
        "destinationIP": "225.0.1.2",
        "sourceIP": "35.1.1.4",
        "outIntfID": [
          "001c.7357.6d61-Ethernet8"
        ]
      }
    ]
  }
],'
```
