Enhancement in Loop Prevention Protocols: Dynamic Per-VLAN Multi-Shortest Path Bridging (DV-MPB)

Abstract. Loops in a network pose significant challenges in modern communication systems, leading to congestion, performance degradation, and potential system failures. Additionally, loops can cause broadcast storms, impacting the network performance. While traditional loop prevention mechanisms like the Spanning Tree Protocol (STP) have been effective, the evolving complexity of network environments demands more efficient solutions. This paper provides a comprehensive review of loop prevention protocols, including STP, Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP), Shortest Path Bridging (SPB), and Per-VLAN Spanning Tree (PVST). Further, this paper introduces Dynamic VLAN - Multi-Shortest Path Bridging (DV-MPB) as a potential solution to enhance loop prevention efficiency. DV-MPB combines the strengths of SPB and PVST to achieve dynamic traffic optimization and reduce convergence time in loop prevention protocols.

Keywords: Shortest Path Bridging · Per VLAN Spanning Tree · Convergence Time · Frame Format · Bridge Per Data Unit

1 Introduction

In networking systems, loops arise when data packets circulate endlessly, failing to reach their intended destinations. These loops pose significant challenges, including network congestion, performance degradation, and potential system failures, particularly within the data link layer where data framing occurs [1]. Network loops introduce various challenges, including broadcast storms, packet loss, duplication, and difficulties in fault isolation, which are further exacerbated by scalability and reliability concerns [1]. As a result, loop prevention becomes a critical aspect of network implementation. While traditional mechanisms such as the Spanning Tree Protocol (STP) [2] were introduced to address such issues, the evolving nature of communication networks necessitates more sophisticated protocols like the Rapid Spanning Tree Protocol (RSTP) [3] and Multiple Spanning Tree Protocol (MSTP) [4].

Shortest Path Bridging (SPB) [5] offers a more efficient and scalable approach to network topology management by dynamically calculating the shortest path through the network using the Intermediate System to Intermediate System (IS-IS) [6] routing protocol. This results in faster convergence, optimal traffic distribution, and improved scalability, which is particularly beneficial for large and complex networks. However, SPB requires a robust understanding of IS-IS and may entail initial configuration complexities. On the other hand, Per-VLAN Spanning Tree (PVST) [7] enhances network performance by creating separate spanning-tree instances for each VLAN, enabling customized loop prevention and path selection. This fine-grained control enhances load balancing and fault tolerance. Nevertheless, PVST introduces increased complexity in network management, potential resource overhead on devices, and compatibility challenges in heterogeneous environments.

Although existing protocols provide effective loop prevention mechanisms, they may suffer from prolonged convergence times, scalability limitations, and limited adaptability to dynamic network conditions. The primary objective of this paper is to tackle the challenge of reducing convergence time in loop prevention protocols, particularly in large-scale and dynamic communication networks where topology changes occur frequently [4]. It's important to focus on developing a solution that minimizes the time required for the network to adapt to topology changes and reconfigure itself to prevent loops, enhancing overall responsiveness and reliability. This paper proposes a loop prevention protocol inspired by the Per-VLAN Spanning Tree (PVST) [7] approach. The Dynamic VLAN - Multi-shortest Path Bridging protocol (DV-MPB) aims to optimize the Per-VLAN Spanning Tree by integrating it with shortest path bridging, ensuring rapid convergence, and maintaining compatibility with existing networking protocols.

The further sections of this paper are structured as follows. A literature survey of existing protocols is presented in the next Section 2, where the drawbacks of existing protocols are discussed. The design and implementation of DV-MPB and the results are presented in Section 3 and 4, respectively. Finally, the paper is concluded in Section 5.

2 Literature Review

This section explores existing loop prevention protocols, analyzing their functionality and effectiveness across diverse network topologies. This review aims to provide valuable perspectives on loop prevention protocols by synthesizing current research, thus enhancing network stability and efficiency.

2.1 Spanning Tree Protocol [2]

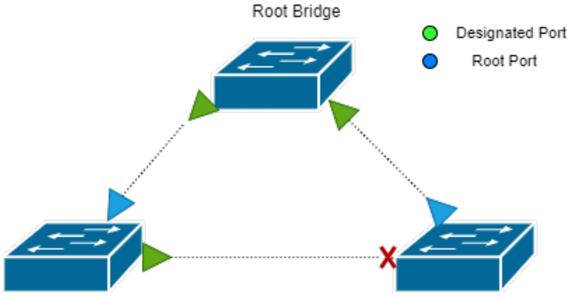

Fig. 1. Spanning Tree Protocol [4]

The Spanning Tree Protocol (STP) is a link layer protocol that prevents loops in bridged Ethernet networks. It was developed in the 1980s and standardized as IEEE 802.1D by the IEEE 802.1 working group [2]. Radia Perlman played a key role in developing STP. Her work included defining the concept of a spanning tree, where redundant links are blocked to ensure a single active path between any two devices in the network. She also created the Bridge Protocol Data Unit (BPDU) [8] format for bridges to exchange information and calculate the best path to the root bridge. STP operates by electing a root bridge within the network based on a combination of bridge priority values and MAC addresses. This root bridge serves as the central point for all traffic flow. Each switch then calculates the shortest path to the root bridge, considering the path cost associated with each link [8]. Ports on switches are placed into different states (forwarding, blocking, or listening/learning) to ensure a loop-free topology. Ports in the forwarding state actively pass data, as shown in the "designated ports" in Fig. 1. Ports in the blocking state prevent loops by blocking traffic, as depicted by the blocked ports in Fig. 1. STP continuously monitors the network topology for changes and dynamically adjusts port states and the topology to maintain loop-free operation. This approach ensures that Ethernet networks remain stable and free from loops, thus preventing issues like broadcast storms and network congestion.

Advantages of STP include automatic configuration, fault tolerance through rerouting, and scalability across network sizes However, it also has limitations such as slow convergence time, the risk of a single point of failure with the root bridge, inefficient bandwidth utilization due to blocked ports, and complexity in configuration and troubleshooting.

2.2 Rapid Spanning Tree Protocol [2]

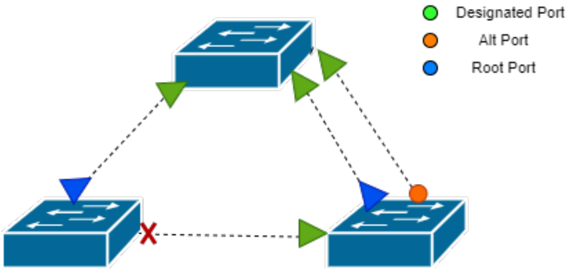

Fig. 2. Rapid Spanning Tree Protocol [4]

Expanding on the Spanning Tree Protocol (STP), the Rapid Spanning Tree Protocol (RSTP) brings significant improvements to network efficiency and how quickly it adapts to changes [9]. Like STP, RSTP was influenced by Radia Perlman, who helped develop it. RSTP speeds up how networks react to changes, thanks to features like quicker port roles and detecting edge ports. It's part of the IEEE 802.1w standard.

While STP focuses on preventing loops by choosing a root bridge and finding the shortest paths, RSTP adds new methods to do this faster. It introduces different port states, like "discarding" and "alternate" ports, to help prevent disruptions during changes in the network as shown in Fig. 2 [3].

RSTP has some clear benefits. It's much faster than STP at reacting to changes, meaning networks can adjust quicker and have less downtime. It also uses network resources more efficiently by making ports switch faster and recovering faster from failures. RSTP handles faults better, quickly getting the network back on track after a problem. But RSTP also has its challenges. It might not work well in networks with RSTP and STP running together [2]. It's also more complicated to set up and fix than STP, and not all devices or setups can handle its fast changes and failover methods.

2.3 Multiple Spanning Tree Protocol [4]

While often credited as the inventor of MSTP, Marshall Eubanks at Cisco spearheaded the initial development, submitting the concept to the IEEE 802.1 working group for further refinement. The resulting IEEE 802.1s standard defines MSTP as an extension of the Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP). Unlike STP and RSTP, which create a single network-wide spanning tree, MSTP allows for multiple instances [4], each tailored to VLANs or VLAN groups, as depicted in Fig. 3.

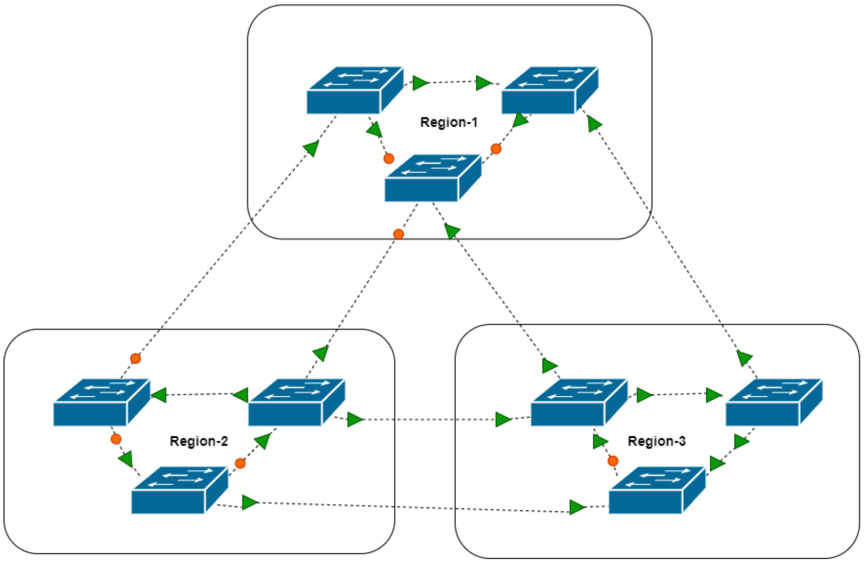

Fig. 3. Multiple Spanning Tree Protocol [4]

This VLAN mapping grants finer control over network topology and resource utilization. By grouping VLANs with similar traffic patterns into separate spanning tree instances, MSTP offers greater flexibility and efficiency in managing network resources.

MSTP offers several advantages. Firstly, it enhances scalability by enabling administrators to create multiple spanning tree instances optimized for specific VLANs [4]. This improves resource utilization and better supports large networks with diverse traffic needs. Secondly, MSTP improves resource utilization by allowing administrators to allocate bandwidth and prioritize traffic based on VLANs, ensuring better performance for critical applications. Finally, MSTP simplifies configuration by grouping VLANs into common spanning tree instances based on traffic characteristics, reducing complexity compared to managing multiple independent spanning trees.

However, MSTP also presents challenges. Compared to STP or RSTP, it introduces additional complexity requiring careful planning and configuration to ensure optimal network performance [4]. Additionally, MSTP may face interoperability issues in heterogeneous environments with devices lacking full support or having different implementations. Finally, creating multiple spanning tree instances can impose extra resource overhead on network devices, requiring careful assessment to ensure sufficient resources are available.

2.4 Shortest Path Bridging [5]

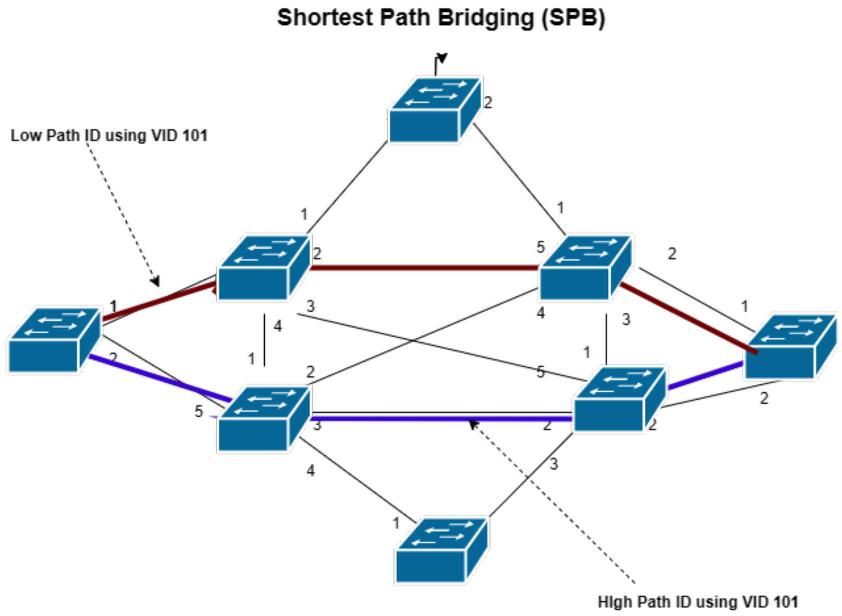
Shortest Path Bridging (SPB)
Low Path ID using VID 101
HIgh Path ID using VID 101

Fig. 4. Shortest Path Bridging [10]

The IEEE 802.1aq Task Group played a pivotal role in defining the initial SPB standard. Researchers at Avaya Labs, particularly Pradeep Palkhiwala and Sanjay Rao, are credited with the initial proposal, which recognized limitations in traditional spanning tree protocols. SPB is a Layer-2 network protocol designed for efficient Ethernet fabric management [10]. Unlike traditional spanning tree protocols, SPB leverages the Intermediate System to Intermediate System (IS-IS) [11] routing protocol to construct Shortest Path Trees (SPTs) [5] for each VLAN, as illustrated in Fig. 4. This approach eliminates complex spanning tree calculations and enables multi-path forwarding, resulting in several key benefits.

SPB offers efficient traffic forwarding by utilizing SPTs to ensure traffic takes the most direct path through the network, minimizing latency and improving overall efficiency. Additionally, SPB's use of IS-IS routing facilitates efficient scaling for large-scale networks with numerous nodes and VLANs, making it suitable for enterprise and data center environments [10]. Finally, SPB provides flexibility by decoupling the logical network topology from the physical one. This allows administrators to adapt to changing network requirements without disrupting existing configurations.

However, implementing SPB can be more complex than traditional STP protocols due to the requirement of a good understanding of IS-IS routing [11]. Additional training or expertise might be necessary for effective deployment and management. Furthermore, SPB may face interoperability challenges in heterogeneous environments where devices lack full support. Compatibility issues between SPB and other protocols could impact network operations and require careful planning to resolve.

2.5 Per-VLAN Spanning Tree [7]

Per-VLAN Spanning Tree (PVST) emerged from collaborative efforts within the networking community, with key figures from the IEEE 802.1 committee playing a vital role [4]. It addresses the limitations of the Spanning Tree Protocol (STP) in VLAN environments by creating separate spanning trees for each VLAN. PVST extends STP principles on a per-VLAN basis. Each VLAN maintains its own spanning tree through switches exchanging Bridge Protocol Data Units (BPDUs) [8] specific to each VLAN. These BPDUs facilitate the election of a root bridge independently for every virtual LAN, as illustrated in Fig. 6.

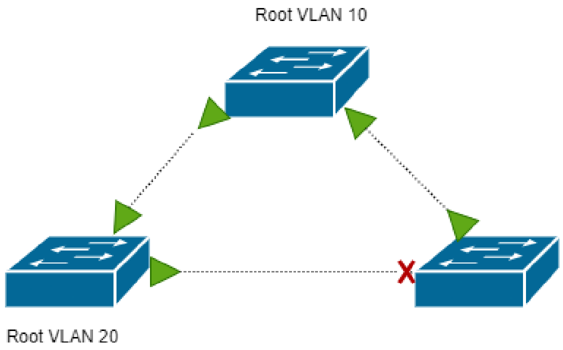

Fig. 5. Per-VLAN Spanning Tree [4]

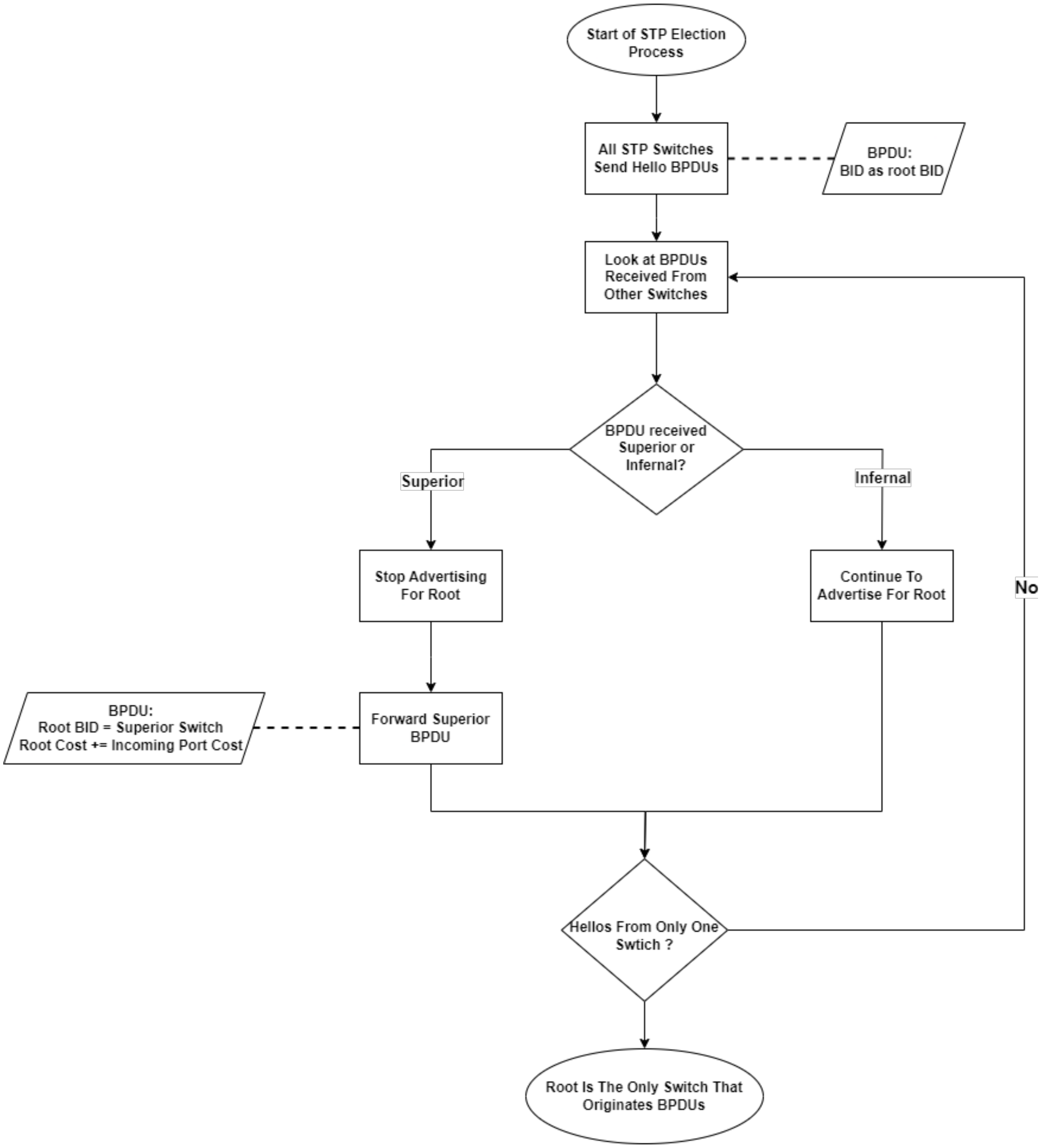
Start of STP Election Process
All STP Switches Send Hello BPDUs
BPDU:
BID as root BID
Look at BPDUs Received From Other Switches
BPDU received Superior or Infernal?
Superior
Infernal
Stop Advertising For Root
Continue To Advertise For Root
No
BPDU:
Root BID = Superior Switch
Root Cost += Incoming Port Cost
Forward Superior BPDU
Hellos From Only One Swtich ?
Root Is The Only Switch That Originates BPDUs

Fig. 6. PVST Flowchart

In PVST, switches begin by sending hello BPDUs to advertise their availability as potential root bridges. These BPDUs include information like the switch's Bridge ID (BID). The switch with the lowest BID is typically elected as the root bridge [8]. It shows the port's transition from blocking to forwarding states. If a superior BPDU is received on a port, the port transitions to blocking. Conversely, if a port receives its own BPDU or a BPDU from a non-superior bridge, it can transition to the forwarding state.

Private VLAN Spanning Tree (PVST) provides advantages such as optimized performance and load balancing by tailoring network topology to each VLAN's needs and faster convergence through independent spanning tree adjustments. However, PVST complexity in implementation and management, resource overhead, and compatibility issues in heterogeneous environments pose challenges [8]. Thus, while PVST improves performance and convergence, its complexities and compatibility concerns require careful consideration in network design and administration.

In conclusion, PVST offers significant performance and convergence benefits over STP in VLAN environments. However, its increased complexity, resource overhead, and potential compatibility issues require careful consideration during deployment.

3 Design and Implementation of DV-MPB

Loop-preventing network protocols encounter challenges with dynamic path selection, relying on spanning trees to identify optimal paths, resulting in inefficient resource utilization and suboptimal performance. This paper introduces Dynamic VLAN-Multi-shortest Path Bridging (DV-MPB) to mitigate these issues. DV-MPB combines the capabilities of SPB (Shortest Path Bridging) and Per-VLAN Spanning Tree (PVST), enabling dynamic traffic optimization.

3.1 Design of IS-IS PDUs in Shortest Path Bridging

Shortest Path Bridging (SPB) utilizes IS-IS (Intermediate System to Intermediate System) Protocol Data Units (PDUs) to facilitate communication and establish shortest paths within the network [5]. IS-IS PDUs, including Link State Protocol Data Units (LSPs) and Sequence Number Packets (SNPs), play a crucial role in SPB's operation. LSPs are used by SPB to exchange link state information between network nodes, enabling them to build a complete view of the network topology [6]. SNPs, on the other hand, are employed to distribute information about the availability of links and nodes, ensuring rapid convergence in case of network changes. By leveraging IS-IS PDUs, SPB efficiently constructs and maintains a dynamic forwarding database, allowing for efficient forwarding of traffic along the shortest paths in the network, thereby optimizing performance and scalability.

IS-IS relies on **Protocol Data Units (PDUs)** for effective communication among routers, crucial for sharing routing information and ensuring network stability. Two of the various PDU types employed by IS-IS are foundational: Hello PDUs and Link State PDUs (LSPs) [6]. Hello, PDUs are instrumental in the initial phase of network discovery, facilitating identifying and establishing neighbor relationships between routers. Meanwhile, Link State PDUs carry detailed network topology information, including updates on link states and reachable destinations within the network. These PDUs are fundamental elements in the functioning of IS-IS, contributing to the robustness and resilience of the routing infrastructure by facilitating efficient communication and accurate dissemination of routing information.

Field Name	Size (Bytes)	Description
PDU Type	1	Identifies the PDU as either an LSPDU (0x83) or an HSPDU (0x84).
Version	1	Indicates the IS-IS protocol version being used.
Reserved	2	Set to zero for future use.
PDU Length	2	Total length of the PDU in octets.
Maximum Area Addresses	1	Maximum number of Area Addresses supported by the router.
Source ID	6	Unique identifier of the sending router.
Checksum	2	Error detection mechanism for the PDU.
PDU-specific Parameters	Variable	Additional information specific to the PDU type (LSPDU or HSPDU).

Table 1. IS-IS Header Frame Format [6]

The **IS-IS header** format is mentioned in Table 1. The PDU Type field distinguishes between Link State PDUs (LSPDU) and Hello State PDUs (HSPDU), with LSPDUs (Type 0x83) conveying link state information and HSPDUs (Type 0x84) managing neighbor discovery and adjacency maintenance. Version denotes the IS-IS protocol version, which ensures seamless compatibility between routers. Reserved bytes, set to zero, are earmarked for future enhancements without current impact on protocol operation [11]. PDU Length specifies the total length of the PDU in octets, which is crucial for accurate parsing by receivers.

Maximum Area Addresses determine the scalability of IS-IS implementations by defining the maximum number of supported Area Addresses. Source ID acts as a unique identifier for the sending router, often incorporating the router's System ID and Area ID for identification within the IS-IS domain. The Checksum field detects errors throughout the PDU, ensuring data integrity during transmission. Finally, PDU-specific Parameters contain additional details specific to the type of PDU, such as network topology and neighboring routers for LSPDUs or neighbor discovery and adjacency establishment for HSPDUs [11].

Field Name	Size(Bytes)	Description
LSP ID	4	Unique identifier of the originating router.
Sequence Number	4	Version control for updates, consistency checks.
Checksum	2	Error detection mechanism for the LSPDU.
Remaining Lifetime	4	Time remaining before the LSP expires and needs to be refreshed.
Type/Length	Variable	Defines the structure and length of TLVs.
TLVs(Type-Length-Value)	Variable	Self-contained information blocks containing details like: 1. Reachable Link TLV: Advertises directly connected links. 2. Network Address TLV: Announces reachable IP prefixes associated with the router.

Table 2. Link State Protocol Data Unit (LSPDU) Format [6]

The **Link State Protocol Data Unit (LSPDU)** format is mentioned in Table 2. The LSP ID field serves as a unique identifier for the originating router of the Link State Protocol Data Unit (LSP), facilitating tracking of routing information sources. Sequence Number enables version control for updates and consistency checks, ensuring synchronization among routers within the network [11]. Checksum provides an error detection mechanism for the LSPDU, safeguarding against data corruption during transmission. Remaining Lifetime indicates the time remaining before the LSP expires and requires refreshing, which is crucial for maintaining up-to-date routing information. Type/Length field defines the structure and length of subsequent Type-Length-Value (TLV) blocks, which encapsulate self-contained information such as reachable links and network addresses, facilitating efficient routing and topology management within the IS-IS network.

Field Name	**Size (Bytes)**	**Description**
Circuit Type	1	Identifies the type of connection (e.g., Point-to-Point, LAN).
Source ID	6	Unique identifier of the sending router.
Holding Time	2	Interval between sending consecutive Hello PDUs.
PDU Length	2	Total length of the Hello PDU in octets.
Neighbor Priority	4	Priority value used for router selection during neighbor establishment.
DIS Flag	1 Bit	Indicates if the router is a Designated Intermediate System within an area.
Authentication Type	Variable	Optional parameters for secure communication (if enabled).
Remaining Parameters	Variable	Additional information specific to Hello PDUs, such as MTU (Maximum Transmission Unit) information.

Table 3. Hello Protocol Data Unit Format [6]

The **Hello Protocol Data Unit (PDU)** format serves as a vital framework for routers to exchange crucial information and establish connections within a network. Each field within the format plays a specific role in facilitating efficient communication and network management. The Circuit Type field identifies the type of connection, such as Point-to-Point or LAN, guiding how routers establish links. Source ID provides a unique identifier for the sending router, ensuring accurate recognition within the network. Holding Time specifies the interval between consecutive Hello PDUs sent by a router, optimizing timing for communication reliability. Meanwhile, the PDU Length field indicates the total size of the Hello PDU in octets, which is crucial for proper data handling and transmission. Neighbor Priority assigns priority values for router selection during neighbor establishment, contributing to efficient network routing and management. Additionally, the DIS Flag indicates whether the router holds a Designated Intermediate System status within the area, influencing network topology and routing decisions. Fields like Authentication Type and Remaining Parameters offer optional and specific information, enhancing security and flexibility in network communications. As illustrated in Table 3, the structured format enables routers to exchange essential data effectively, supporting optimal network performance and configuration.

3.2 Design of BPDUs in Per-Vlan Spanning Tree

In PVST (Per-VLAN Spanning Tree), Bridge Protocol Data Units (BPDUs) are tailored for individual VLANs, ensuring optimal network stability and loop prevention. These VLAN-specific BPDUs carry essential information such as Bridge IDs and Path Costs, allowing PVST to maintain separate spanning trees for each VLAN. By utilizing VLAN-specific BPDUs, PVST can identify and block redundant paths on a per-VLAN basis, enhancing network performance and redundancy. Moreover, BPDUs facilitate rapid convergence by propagating topology changes within specific VLANs, leading to faster response times than traditional STPs. Overall, using VLAN-specific BPDUs in PVST optimizes network efficiency and fault tolerance while minimizing the risk of loops within individual VLANs.

In PVST, BPDUs are used to manage spanning tree information separately for each VLAN in the network. These BPDUs carry specific details like VLAN ID, bridge priority, and path cost, helping switches create distinct spanning trees for every VLAN. By sharing these BPDUs, switches determine the root bridge for each VLAN, prevent loops, and quickly adapt to network changes. The format of BPDU is mentioned in Table 4 as given below.

Field Name	**Size (Bytes)**	**Description**
Protocol ID	2	Always set to 0x0000 (IEEE 802.1D standard)
Version ID	1	Usually 0x00 for PVST (original STP version)
BPDU Type	1	0x00: Configuration BPDU (most common) 0x80: Topology Change Notification (TCN) BPDU
Root Bridge ID	8	Bridge with the lowest BID (root for spanning tree) - BID = Bridge Priority (2 bytes) + Switch MAC (6 bytes)
Root Path Cost	4	Cumulative cost to reach the root bridge from sending switch
Bridge ID	8	BID of the switch sending the BPDU - Same structure as Root Bridge ID
Port ID	2	Switch port on which the BPDU is transmitted/received
Message Age	2	Time elapsed since root bridge sent original BPDU
Max Age	2	Maximum time to wait for BPDU from root bridge before recalculation
Hello Time	2	Interval for sending Configuration BPDUs to maintain communication between switches
Forward Delay	2	Time to wait after port transition to designated port before forwarding

Table 4. BPDU Frame Format

when comparing the BPDUs, the election of the root bridge is decided by the comparison of the following parts of the BPDUs sent and received by the bridge in the order,

- Root Bridge ID
- Root Path Cost
- Bridge ID
- Port ID

The other values, such as Message Age, Max Age, and other parameters, are involved in the later consideration of changed BPDUs over time. Such an example involves when the BPDU received has a different Root Bridge ID from the current Root Bridge ID. For example, if a received BPDU contains a different Root Bridge ID than the current one, the switch evaluates the Message Age to determine the BPDU's timeliness. It verifies whether the Message Age exceeds the Max-Age parameter, indicating potential staleness and the switch initiates the re-convergence process for the affected VLAN. This approach ensures dynamic adaptation to network changes, bolstering stability and performance. Considering these additional parameters, PVST reinforces network resilience and responsiveness, solidifying its effectiveness in spanning tree management.

3.3 Design of Modified BPDU in DV-MPB

DV-MPB uses a modified version of the standard BPDU to leverage the benefits of Source Path Bridging (SPB) within a VLAN. This means it keeps the existing information carried by BPDUs in traditional protocols (like PVST) but adds a new field called "SPB Weight". This field is flexible in size. It can be:

- 2 bytes: This is a more compact format. It specifies if SPB is enabled/disabled for the root bridge ID and provides a general weight for all paths in the SPB instance.
- 4 bytes: This offers a more detailed representation. It can hold the complete root bridge ID or a combination of the designated bridge ID and root port ID. Additionally, it provides the weights for all paths in the SPB instance.

Adopting this format helps update the Shortest Path Table (SPT) and, subsequently, the Active Path Table (APT) once these data units are transmitted. It facilitates the election of an SPB Instance upon receipt of this packet by the VLAN. Furthermore, it assists in updating the sending data unit following further network updates. This format aids in analyzing the weights of the paths selected in the SPB, facilitating loop prevention. By advertising path weights, switches can collectively determine the optimal paths within the SPT while avoiding loops. Analyzing these weights helps network administrators understand the costs associated with each path and identify potential loops.

Furthermore, the BPDU format facilitates continuous updates. As network conditions change (i.e., link failures), switches can modify their SPB weight values and propagate these updates through subsequent BPDUs. This ensures all switches have the latest information to maintain an accurate and up-to-date SPT and APT. Various test cases were simulated using the described frame format and the IS-IS protocol. These simulations produced SPB instances and transmitted modified BPDUs across all VLANs, ensuring a loop-free network configuration.

Root Bridge Election:

- Loop Prevention within each VLAN: A SPB instance is deployed in each VLAN; SPB ensures loop-free topologies by using the shortest path calculation via Dijkstra's algorithm and by blocking redundant links to prevent loops in each VLAN.
- The root bridge of the VLAN is taken based on the bridge that reaches out on a maximum number of paths in VLAN.
- Election of the Root Bridge of the whole network of VLANs is done by comparing the Modified BPDUs. The comparison of the following parts of the Modified BPDUs sent and received by the bridge in the order, Root Bridge ID, Root Path Cost, SPB weight, Bridge ID, and Port ID.
- The rest part of the election remains the same as the PVST.

By following these steps, SPB constructs a loop-free network topology within the VLAN, electing the root bridge at its center, ensuring efficient and reliable data transmission using the shortest paths in the network. The further part of this section explains the way the proposed solution configures the whole network.

3.4 Implementation of DV-MPB

DV-MPB operates continuously, adapting to network changes in real-time. At its core is a modified Shortest Path Bridging (SPB) algorithm, which calculates the shortest path for each node and VLAN. The algorithm considers link weights, congestion and complexity, and VLAN information. All paths and weights of this nature are updated in the Shortest Path Table (SPT) for each VLAN.

Additionally, DV-MPB incorporates dynamic reconfiguration capabilities to respond to network topology alterations or failures swiftly. When network conditions change, DV-MPB initiates rapid recalculations of the shortest paths for affected VLANs, ensuring optimal routing efficiency while maintaining seamless connectivity. This adaptability is crucial in modern networking environments where agility and resilience are paramount. By continuously monitoring and adjusting network paths based on real-time data, DV-MPB enhances overall network performance and reliability.

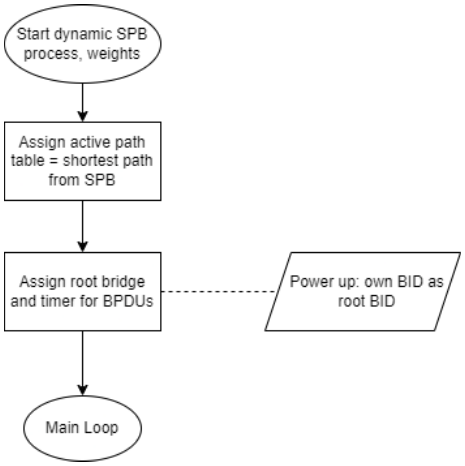
Start dynamic SPB process, weights
Assign active path table = shortest path from SPB
Assign root bridge and timer for BPDUs
Power up: own BID as root BID
Main Loop

Fig. 7. DV-MPB: SPB Election

Fig. 7 explains the election of SPB in each VLAN. Firstly, It starts by setting the initialization of all VLANs; in each VLAN, it sets up an SPB instance, the SPB-weights (Eq. 1), tables, and sets the root BIDs to the VLAN's own BIDs.

$$SPB_{weight} = \sum_{s,t \in V, s \neq t} \frac{d(s,t)}{n(n-1)} \tag{1}$$

Where, SPB_{weight}: The average distance between all pairs of vertices in the graph, V: The set of vertices in the graph, $d(s,t)$: The shortest path distance (often measured by the number of edges) between two vertices s and t in the graph G, n: The number of vertices in the graph.

The Eq. 1 is used in the analysis to quantify the average distance between nodes in a network or graph. It measures how well-connected or close the nodes are to each other in the network. This is directly calculated from the SPT of the VLAN.

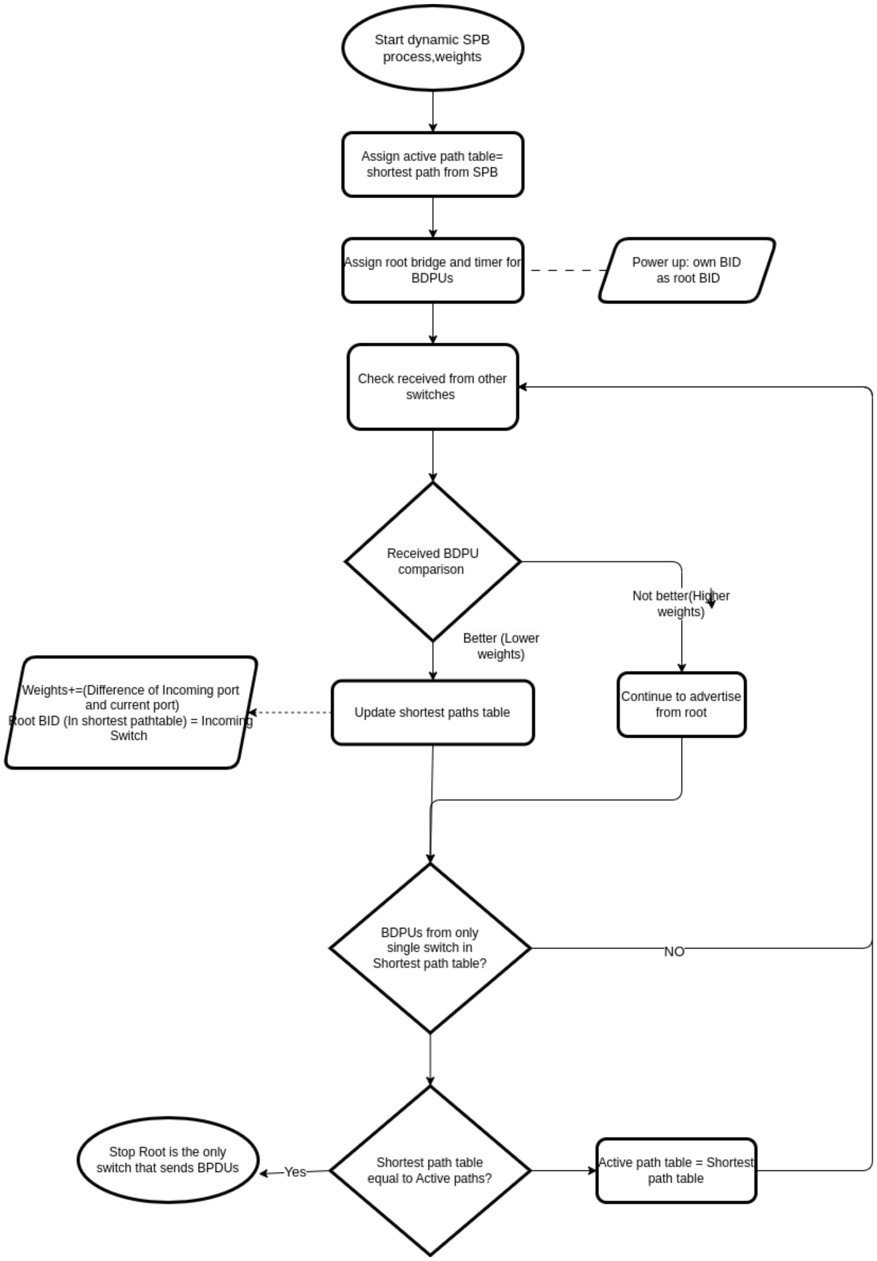
Start dynamic SPB process,weights
Assign active path table= shortest path from SPB
Assign root bridge and timer for BDPUs
Power up: own BID as root BID
Check received from other switches
Received BDPU comparison
Not better(Higher weights)
Better (Lower weights)
Weights+=(Difference of Incoming port and current port)
Root BID (In shortest pathtable) = Incoming Switch
Update shortest paths table
Continue to advertise from root
BDPUs from only single switch in Shortest path table?
NO
Stop Root is the only switch that sends BPDUs
Yes
Shortest path table equal to Active paths?
Active path table = Shortest path table

Fig. 8. Dynamic VLAN- Multi-shortest Path Bridging

Fig. 8 explains the election of the root bridge for the whole network. The modified BPDUs are sent over the network along each port. Once the modified BDPUs are received by a switch, it performs a check of weights. If the weights, i.e., SPB-weight and the port weight, are lower than the previous ones in the switch, leading to the received modified BPDU being superior, this changes the root BID of the current bridge. These updates are then reflected in the SPT and forwarded to another check to verify the origin of BPDUs from only one switch. If not, the BPDUs are checked again and changed until only one switch originates BPDUs.

Next, DV-MPB selects a subset of calculated paths from the Shortest Path Table (SPT) for each VLAN, considering factors such as load balancing, congestion awareness, and path diversity. These selected paths are stored in the Active Path Table (APT) and programmed into forwarding tables to ensure efficient packet delivery. Notably, DV-MPB employs separate shortest paths for each VLAN, derived from the active paths specific to that VLAN stored in the APT. This approach directs only relevant traffic flows within each VLAN, maximizing efficiency. Furthermore, inspiring by the concept of PVST, where separate trees are maintained for each network. DV-MPB adopted a similar strategy by configuring switches to handle separate active paths for each VLAN.

Once the table is updated, the APT will be compared. If the tables aren't the same, the APT is updated to the currently calculated SPT. When the two are equal, it has a DV-MPB with only one switch originating BDPUs. Every VLAN is continuously optimized by a dynamic loop of a modified SPB, which updates the SPT and the APT to provide us with better VLANs when the network within each VLAN changes. This also helps reduce reconfiguration time as the APT changes as the network changes within a VLAN.

4 Result and Analysis

The proposed approach involved implementing simulations by creating several VLANs in a network and instances of SPB for each VLAN. The root node was elected over these instances via the SPT table. Additionally, a set of VLANs was created, resulting in the transmission of Modified BPDUs over the Root Switches of the network. The core of the implementation utilized protocol logic and the Networkx Python library for network setup, control, and visualization [12]. The simulations primarily focused on evaluating the performance of PVST (Per VLAN Spanning Tree) and SPB (Shortest Path Bridging) in a multi-VLAN network simulated environment.

The evaluation of the proposed method utilized a system with a 12th Gen Intel® Core™ i7-12650H processor featuring 16 cores and 16 GB of RAM. The average convergence time was calculated for various network sizes with different numbers of VLANs. To achieve this, diverse random networks were generated, ranging from 3 to 60 VLANs, each with distinct SPB Instances for every VLAN and random port connection. The average convergence time was also computed for 10 different random network instances generated by the simulation.

Using the above frame format, i.e., the modified BPDU and the IS-IS protocol, several test cases were simulated, producing instances of SPBs for each VLAN for the DV-MPB and transmission of the above format data units overall VLANs and producing the whole network with no loops. Here is a case of a set of VLANs; the proposed solution sets all VLANs, deploying the SPB instances over all the VLANs and sending the modified BPDUs over all VLANs. Therefore, the DV-MPB approach leads to the following configuration of the Network:

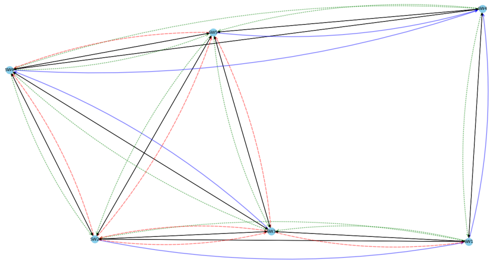

Fig. 9. PVST over small network

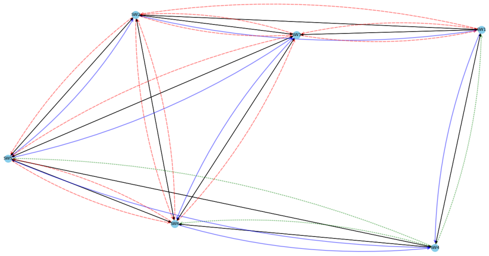

Fig. 10. DV-MPB over small network

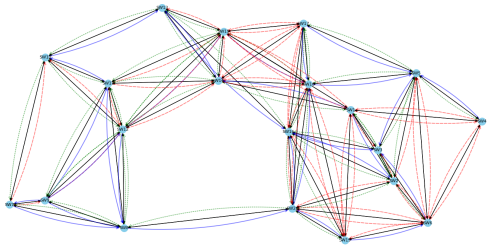

Fig. 11. PVST over large network

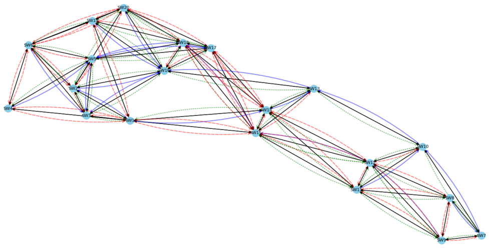

Fig. 12. DV-MPB over large network

In Fig. 9, 10, 11, 12 the blue straight lines indicate that the port is a Root Port, the dotted lines indicate that the port is a Designated Port and is in forwarding state, and the dashed lines indicate that the port is an Undesignated Port and is in blocked state. These Fig. 10, 12 indicate the Modified BPDUs converge into a single root and leading no loops in the network. These are similar to the results obtained via the PVST approach Fig. 9, 11. Also, these figures of the simulations give the output of the first convergence of all the modified BPDUs and modified BPDUs for DV-MPB and PVST, respectively. The existence of loops in the network is absent after the convergence of BPDUS. Further convergences of BPDUs help discover all VLANs and ports in the network. The above simulations indicate that the DV-MPB works for loop prevention in the network. Moreover, the comparison of DV-MPB and PVST is shown in Fig. 9 and Fig. 10 shows the comparison between PVST and DV-MPB for a small network of 6 VLANs. And Fig. 11 and Fig. 12 show the comparison for a large network of 20 VLANs, indicating the scalability of DV-MPB.

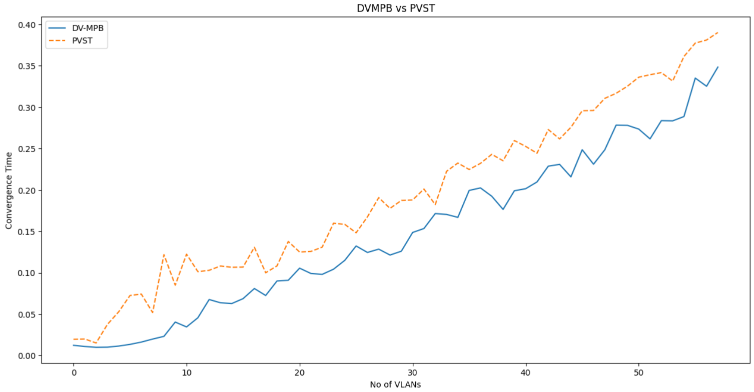

Fig. 13. DV-MPB And PVST: Convergence Time vs No. of VLAN's

On analysis of Fig. 13, it shows that the Convergence time of DV-MPB (Solid Line) is faster than the PVST (Dotted Line), clearly showing better signs converging faster in large scale networks. This proves the better convergence times compared to PVST with the scalability of networks. The major reason is that the SPB instances per VLAN have faster convergence than STP instances per VLAN in PVST. Also, in the modified BPDU comparison, a major role is played by the SPB weights during the first few iterations, as the Root Path Cost is initiated as zero, whereas the SPB weights remain a constant value for comparing as long as the network doesn't change while BPDUs converge. On the other hand, the Root Path Cost adds up as the network is being discovered by the BPDUs.

5 Conclusion and Future works

This paper's proposed approach, DV-MPB, yielded interesting results in preventing loops in multi-VLAN networks. The simulation focused on a network with multiple VLANs and implemented a method for sending modified BPDUs across root switches. The performance of two spanning-tree protocols, PVST and SPB, was then analyzed. The simulation results in this paper were simulated over several hundred times for various constructed networks and converged in all cases used for the simulation. The analysis confirmed that PVST achieves high efficiency in data flow for smaller networks. However, the scalability of our proposed DV-MPB, especially in very large networks. The simulations also validated the effectiveness of combining network segmentation with SPB. This approach ensures efficient path calculation within each VLAN and prevents loops via a successful root switch election and the formation of the Shortest Path Tree (SPT) for each VLAN.

Looking ahead, several areas may require further exploration. It is needed to evaluate the approach to identify and address any potential vulnerabilities. Additionally, investigation is necessary to integrate the solution into the network layer for a more robust implementation. This might involve modifying existing protocols or developing a new one entirely. Finally, future research will focus on optimizing calculations of the Shortest Path Table and root switch selection algorithms, potentially through more sophisticated algorithms or even machine learning techniques.

References

1. P. Shyam, Network loops and loop avoidance , https://www.linkedin.com/pulse/network-loops-loop-avoidance-priyanka-kumari (2020).
2. W. Wojdak, Rapid Spanning Tree Protocol: A new solution from an old technology, http://pdf.cloud.opensystemsmedia.com/xtca-systems.com/PerfTech.Mar03.pdf (2003).
3. A. Inc., Rapid Spanning Tree Protocol (RSTP), https://www.accuenergy.com/support/reference-directory/rapid-spanning-tree-protocol-rstp/ (2024).
4. D. A. O. H. A. Rawyer Asaad Rashid, Dana Faiq Abd, Performance Evaluation using Spanning Tree Protocol, Rapid Spanning Tree Protocol, Per-VLAN Spanning Tree, and Multiple Spanning Tree, https://journals.uhd.edu.iq/index.php/uhdjst/article/view/1254/841 (2024).
5. E. David Allan, Shortest Path Bridging: Efficient Control of Larger Ethernet Networks, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5594687 (2010).
6. D. A. E. N. B. P. U. A. D. Fedyk, Peter Ashwood-Smith, IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging, https://www.rfc-editor.org/rfc/pdfrfc/rfc6329.txt.pdf (2012).
7. S. Kasu, Spanning Tree Protocol, https://dspace.sunyconnect.suny.edu/server/api/core/bitstreams/13b14afb-50c1-4264-a9f3-6ee0421a2487/content (2015).
8. Y. Indrianingsih, Spanning tree protocol (stp) based computer network performance analysis on bpdu config attacks and take over root bridge using the linear regression method, https://join.if.uinsgd.ac.id/index.php/join/article/view/703 (2021).
9. P. Rathod, Comparative Analysis of Spanning Tree Protocol and Rapid Spanning Tree Protocol using Packet Tracer, https://soe.rku.ac.in/conferences/data/75_8556_ICSET%202022.pdf (2022).
10. P. Ashwood-Smith, Shortest Path Bridging IEEE 802.1aq Overview, https://picture.iczhiku.com/resource/paper/wHKsUgyUHEyAWVcm.pdf (2011).
11. Y. K. S. K. Mijeong Yang, Jinho Hahm, Design and Implementation of the IS-IS Routing Protocol with Traffic Engineering, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1274270 (2004).
12. Networkx-Network Analysis in Python, https://networkx.org/ (2024).