IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Review of Computer Engineering Studies
Vol. 11, No. 4, November, 2024, pp. 53-63
Journal homepage: http://iieta.org/journals/rces

Technical Comparison of FANS Models: Q-in-Q, VXLAN and MPLS in Shared Networks

Guido Priano[1], Daniel Alberto Priano[2], Javier Guevara[2], Matías Marsicano[2], Fabio Sergio Bruschetti[2], Isabella Anguillesi[3], María Claudia Abeledo[2*]

[1] Facultad Regional Buenos Aires (FRBA), Universidad Tecnológica Nacional (UTN), Ciudad Autónoma de Buenos Aires 1179, Argentina
[2] Centro de Investigación y Desarrollos en Informática (CIDI), Instituto de Tecnologías Emergentes y Ciencias Aplicadas (ITECA), Escuela de Ciencia y Tecnología (ECyT), Universidad Nacional de San Martín (UNSAM), Ciudad de San Martín, Provincia de Buenos Aires 1650, Argentina
[3] Escuela de Ciencia y Tecnología (ECyT), Universidad Nacional de San Martín (UNSAM), Ciudad de San Martín, Provincia de Buenos Aires 1650, Argentina

Corresponding Author Email: mabeledo@unsam.edu.ar

https://doi.org/10.18280/rces.110401

Received: 15 October 2024
Revised: 11 November 2024
Accepted: 20 November 2024
Available online: 25 November 2024

Keywords:
5G, FANS, GPON, IoT, MPLS, Q-in-Q, shared networks, VXLAN

ABSTRACT

This article presents a technical comparison of the three principal interconnection models defined in the Fixed Access Network Sharing (*FANS*) standard. The three models are *Q-in-Q*, *VXLAN* and *MPLS*. This standard has been developed by the Broadband Forum as a solution to enable multiple operators to share fiber optic infrastructures efficiently, which is a crucial aspect in the deployment of *5G* and *IoT* networks. Each model is evaluated in terms of its scalability, complexity of deployment, capacity for traffic isolation, and suitability for specific use cases. *Q-in-Q* offers simplicity and compatibility with Ethernet networks, rendering it an optimal choice for them. *VXLAN* is distinguished by its remarkable scalability in virtualised environments and data centres. *MPLS*, on the other hand, is particularly suited to networks that necessitate the handling of diverse traffic types with guaranteed Quality of Service (*QoS*). It can be surmised from this comparison that the decision regarding the most appropriate model will depend on both the specific requirements of the operators in question and the challenges they face in integrating new technologies. The implementation of these solutions has the potential to reduce operational costs and enhance the efficiency with which network resources are utilised.

1. INTRODUCTION

In recent years, the growing demand for bandwidth, driven by the proliferation of high-definition video and the Internet of Things (*IoT*), has prompted the telecommunications industry to develop fiber optic infrastructure in closer proximity to the end user (*FTTH* - Fiber to the Home) with the objective of enhancing the speed of services [1]. This has resulted in significant capital expenditure and a number of challenges for operators, who are seeking solutions to enable the efficient sharing of these infrastructures [2].

In response to this need, the concept of *FANS* (Fixed Access Network Sharing) was developed, which allows multiple operators to share a single access network [3]. The Broadband Forum standardised *FANS* through the TR-370 standard, with the objective of offering flexible and automated management for shared networks. This facilitates the interconnection of infrastructure providers (*InPs*) and wholesale operators, or Virtual Network Operators (*VNOs*). Consequently, *FANS* permits the logical partitioning and isolation of shared network resources between operators, and is compatible with virtualisation, whereby control functions are migrated from dedicated network equipment to software running on generic hardware. *FANS* also provides Network as a Service (*NaaS*).

This study is specifically focused on fibre optic access networks, with particular emphasis on *PON* (Passive Optical Networks), and employs *GPON* (Gigabit *PON*) as the primary technology for analysis. *GPON* technology is founded upon the ITU-T G.984.1 standards [4]. The aforementioned standards elucidate the regulatory framework pertaining to the provisioning, protocols, maintenance, fibre installations, privacy and security. In a *GPON* network, a client concentrator, designated as an *OLT* (optical line termination), serves to interconnect disparate clients through the use of *ONT* (optical network terminal) terminals. The *OLTs* are designed with the capacity to serve up to 128 customers per fibre port. In order to achieve this capacity, each fibre optic strand is split with splitters that separate the signal and send it from one input port to multiple output ports.

In this context, the role of *IoT* devices is of particular significance, as they represent a key component of the most prominent applications of *5G* networks, as defined by ITU-R [5], called *mMTC* (Massive Machine Type Communication). It is imperative that these networks are capable of supporting the anticipated billions of devices, including smart home appliances, industrial sensors, and emergency systems, all of which will be connected through low latency and high reliability technologies. This article will analyse three main interconnection models within the *FANS* standard. This article compares and contrasts the characteristics and technical capabilities of three interconnection models: *Q-in-Q*, *VXLAN*, and *MPLS*.

Clearly, these are essential technologies used in modern networks to improve traffic management, scalability and security. Each serves a different purpose, but they can also be integrated to create robust network architectures.

The following is a detailed comparison of these technologies, highlighting their functionalities, benefits and use cases, in the following order: First, an analysis of each of them is provided. Secondly, their advantages and limitations are explained. Thirdly, a comparative synthesis is made, in accordance with the work and experiments carried out in the IPlan company [6]. Finally, conclusions are drawn.

2. Q-IN-Q MODEL

The *Q-in-Q* model, also referred to as provider bridging or stacked VLANs, has been formalised as IEEE 802.1ad [7], an Ethernet networking standard that was incorporated into the 1998 IEEE 802.1Q standard in 2011. The original specification permitted the insertion of a single *VLAN* header into an Ethernet frame; however, this technology enables the encapsulation of multiple *VLAN* tags within a single frame. Together, they form a tag stack, which facilitates the creation of Metro Ethernet networks that can efficiently handle large amounts of traffic and multiple carriers. The term '*VLAN* tag' is typically employed to denote the 802.1Q *VLAN* header in a streamlined form. In the context of an Ethernet frame, a *Q-in-Q* frame is defined as a frame that has two 802.1Q *VLAN* headers, henceforth referred to as dual-tagged. The *Q-in-Q* scheme is shown in Figure 1.

The 802.1ad standard, published in 2006, delineates the architectural framework and communication protocols for the provision of discrete instances of media access control (*MAC*) services to multiple independent users on a bridged local area network (*LAN*). This is achieved in a manner that does not necessitate collaboration between users and with minimal interaction between users and the *MAC* service provider. The objective is to afford customers the option of operating their own *VLANs* within the *VLAN* provided by the service provider. In this manner, the service provider is able to configure a *VLAN* for the customer, who is then able to treat that *VLAN* as if it were a trunk.

The necessity for this standard arises from the constraints inherent in its predecessor. The primary limitation of 802.1Q is its 12-bit *VLAN ID* field, which allows for a maximum of 4,096 tags. However, with the use of double tagging, the number of tags that can be accommodated reaches 16,777,216, which is sufficient for modern networks. The introduction of a second tag enables the execution of operations that would otherwise be inaccessible if the *VLAN ID* field were to be expanded from 12 bits to 24 bits or more. The presence of multiple tags allows for the modification of frames by switches, including the addition, deletion, or modification of tags.

Furthermore, a frame comprising multiple tags not only possesses multiple *VLAN IDs* but also multiple *VLAN* header bit fields. The tag stack provides an effective mechanism for Internet Service Providers (*ISPs*) to encapsulate customer single-tagged 802.1Q traffic with a single tag, resulting in a *Q-in-Q* frame at the end. The outer tag is employed for the purpose of identifying and segregating traffic from disparate clients, while the inner tag is retained from the original frame.

The creation of layer 2 (*L2*) tunnels and the application of Quality of Service (*QoS*) policies are both enabled by *Q-in-Q* frames. Furthermore, it is backward compatible with 802.1Q. Although 802.1ad is limited to two tags, there is no maximum limit in the standard that restricts a frame to a maximum of two tags. This allows for future growth of the protocol, as evidenced by the fact that service provider topologies utilise frames with more than two tags.

On the other hand, it is more straightforward for network equipment manufacturers to modify their equipment by creating multiple 802.1Q headers than to modify their equipment to implement a new extended *VLAN ID* field header that is not 802.1Q-compliant.

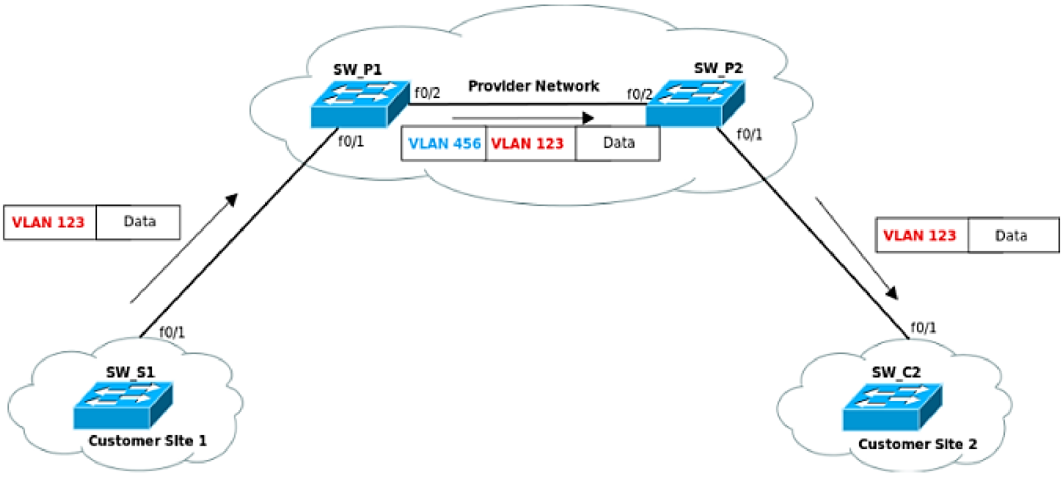

Figure 1. *Q-in-Q* scheme

The encapsulation is then analysed in this model according to Figure 2. An 802.1Q header, comprising four bytes, is appended to an untagged Ethernet frame as follows: the tag is inserted between the source *MAC* address (*SAMAC*) of the untagged frame and its ethertype field. The newly inserted *VLAN* header's ethertype is set to 0×8100, thereby identifying the following data as a *VLAN* tag. A total of 12 bits are allocated for the *VLAN ID*, while the remaining bits within the *VLAN* fields are populated in accordance with the specific policy (e.g., *QoS*) associated with the interface on which the tag imposition occurred. Following the insertion of an 802.1Q header into an untagged frame, the original ethertype of the frame appears to have been altered to 0×8100. The original ethertype of the untagged frame (in the single tag frame) is now in close proximity to the data (payload), yet its value remains unaltered.

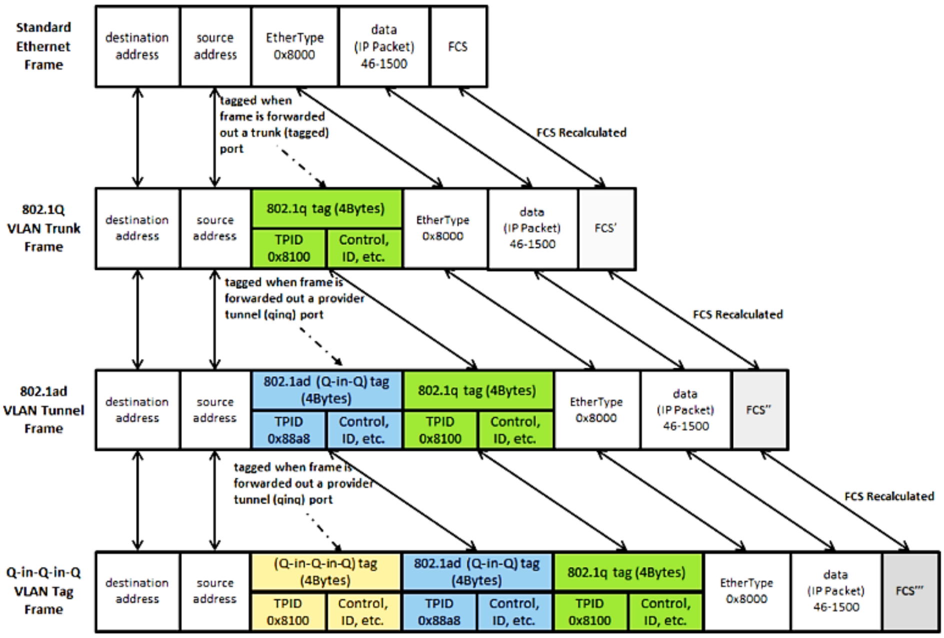
Standard Ethernet Frame
destination address
source address
EtherType 0x8000
data (IP Packet) 46-1500
FCS
tagged when frame is forwarded out a trunk (tagged) port
FCS Recalculated
802.1Q VLAN Trunk Frame
802.1q tag (4Bytes)
TPID 0x8100
Control, ID, etc.
FCS'
tagged when frame is forwarded out a provider tunnel (qinq) port
802.1ad VLAN Tunnel Frame
802.1ad (Q-in-Q) tag (4Bytes)
TPID 0x88a8
FCS''
Q-in-Q-in-Q VLAN Tag Frame
(Q-in-Q-in-Q) tag (4Bytes)
FCS'''

Figure 2. *Q-in-Q* encapsulation [3]

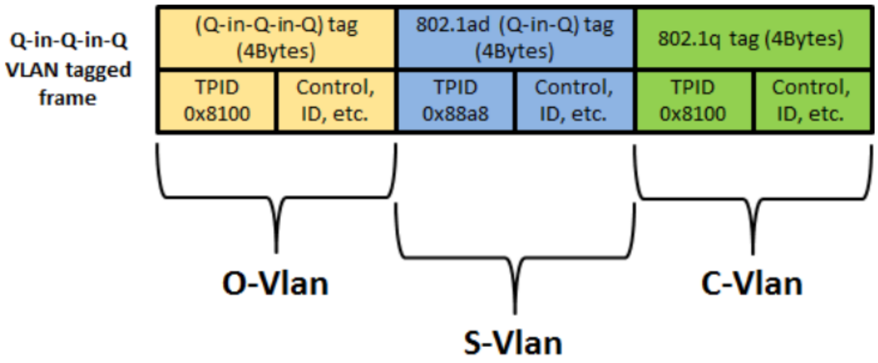

Figure 3. Operator, client and service *VLAN* scheme [3]

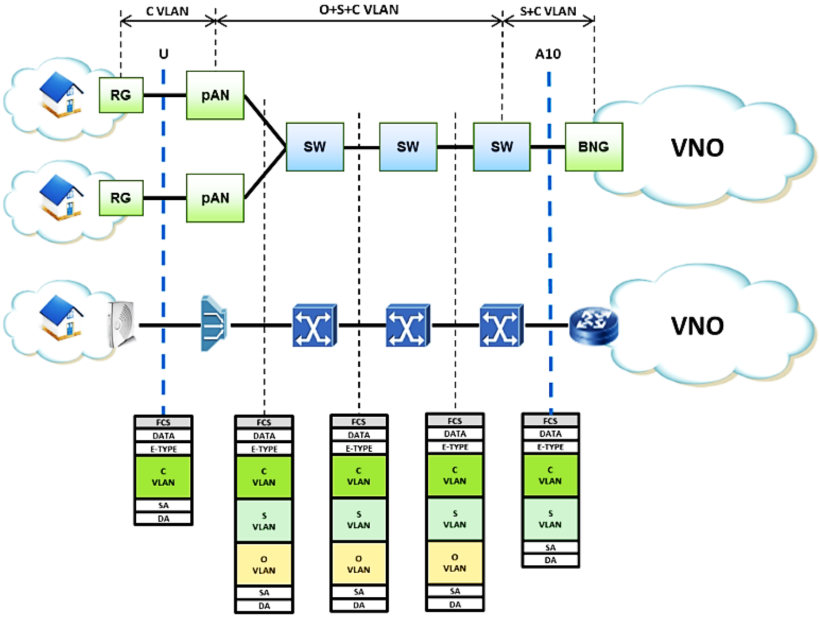

Figure 4. *Q-in-Q* scheme in *FANS* [3]

Subsequently, a second 802.1Q header is appended to a single tag frame in the following manner: the second tag is inserted in front of the first tag, which is thus situated in closer proximity to the Ethernet header than the original tag. The second tag is then inserted between the *MAC SAMAC* and the first tag. The second tag is assigned the default ethertype of 0×88A8, which differs from the standard 802.1Q 0×8100. A total of 12 bits are allocated for the *VLAN ID*, while the remaining bits within the *VLAN* fields are populated in accordance with the policy of the interface on which the tag imposition occurred.

The insertion of a third tag will result in its placement in front of the previous tags, in closer proximity to the Ethernet header. The original, unencoded Ethernet type of the frame is always located after all tags and adjacent to the data. In the case of an 802.3 frame, the aforementioned ethertype would be a 'length' value, denoting the length of the frame up to the end point. In the case of an 802.3 frame with an *LLC* header, the *LLC* header remains after the length field and is adjacent to the data.

Finally, in a tag stack, the push and pop operations are performed at the outer end of the tag. Consequently, the tag added by an insert operation becomes a new outer tag, while the tag to be removed is the current outer tag.

2.1 Interconnection of Q-in-Q model

The *Q-in-Q* model presents a network topology comprising Service *VLANs* (*S-VLANs*) and Client *VLANs* (*C-VLANs*) for each provider. The aforementioned *VLANs* are employed so that the client is able to utilise any *VLAN*, provided that it is associated with an *S-VLAN* that is specific to that particular provider. In order to facilitate the implementation of *VLANs* in the context of *FANS*, it is proposed that a new Operator *VLAN* tag, or 'Operator *VLAN'* (*O-VLAN*), be introduced as a third *VLAN* tag, in addition to the existing C-VLAN and *S-VLAN*. This enables the *VNO* to oversee two *VLAN* levels (*S + C VLAN*) for its service configurations, whereas the *InP* merely assigns the O-Tag for each *VNO*.

In this configuration (Figure 3), the *ONTs* of various *ISPs* may share the same *VLAN* on their uplink. This *VLAN* is then encapsulated within an *S-VLAN* and transmitted to the provider's network.

As illustrated in the Figure 4, the *C-VLAN* tag information is transmitted across the network. In the downstream direction, the carrier *O-VLAN* information is incorporated into the Ethernet frame at the adjacent switch, situated at the reference point (A10). The aforementioned information tags remain intact until they reach the physical access node (*pAN*). In contrast, in the upstream direction, the *S-VLAN* and *O-VLAN* tag information is incorporated into the *C-VLAN* tag within the Ethernet frame at the *pAN*. It is crucial to highlight that the *O-VLAN* data is discarded on the switch situated in proximity to the A10 reference point, whereas the *S-VLAN* information persists in traversing the *VNO* network.

2.2 Advantages of Q-in-Q

- Scalability: The use of additional tags (*O-VLANs*) allows *Q-in-Q* to support a large number of *VLANs*, overcoming the limitations of the original IEEE 802.1Q standard.
- Traffic isolation: Encapsulation of multiple *VLAN* tags ensures that traffic from different carriers remains completely isolated, increasing security in shared networks.
- Compatibility: *Q-in-Q* is compatible with existing Ethernet networks, making it easy to deploy in networks already using Ethernet technologies1.
- Traffic segregation: The provider [6] can effectively segregate customer traffic using outer *VLAN* tags (Service *VLAN ID*) while preserving the inner *VLAN* tags (Customer *VLAN ID*). This isolation ensures that customer data remains secure and separate from other customers' traffic [8-10].
- Increased *VLAN* capacity: By implementing *Q-in-Q*, the provider [6] can extend the number of available *VLANs* from 4096 to over 16 million, accommodating a large customer base without requiring unique *VLAN* ranges for each customer.

2.3 Limitations and technical challenges of Q-in-Q

- Processing overhead: The use of multiple tags can create additional overhead on network equipment, especially in large-scale networks.
- Equipment compatibility: Not all network devices support the *Q-in-Q* standard, which may limit its deployment in older infrastructures.
- Limited scalability: Despite its greater capacity compared to traditional *VLANs*, *Q-in-Q* remains less scalable than newer technologies such as *VXLAN*, as explained in the next section.
- Complexity in management: While *Q-in-Q* simplifies *VLAN* management by allowing overlapping *IDs*, it can introduce complexity in configurations, especially as the number of customers grows [6]. Service providers must ensure that both inner and outer tags are correctly assigned and maintained [11, 12].
- Performance impact: In scenarios with extensive use of double tagging, there may be performance impacts due to additional processing required for handling extra tag information in each frame [12].

3. VXLAN MODEL

The *VXLAN* (Virtual Extensible Local Area Network) model represents a technological advancement that enables the establishment of overlay networks over existing physical networks. The *MAC* Address-in-User Datagram Protocol (*MAC-in-UDP*) is employed to encapsulate data link layer traffic at the network layer, thereby enabling the transmission of Ethernet frames over an Internet Protocol (*IP*) network. The original intention was to provide the same services as a traditional *VLAN*, albeit with limited extensibility and flexibility. The *IETF* has standardised it in *RFC* 7348 [13], which is currently in the Informational status category.

The advent of *VXLAN* can be attributed to the proposals put forth by various manufacturers of networking equipment and other technologies, including Cisco, Arista Networks, Broadcom, Intel, *VMware*, and others. These proposals were made in an effort to circumvent some of the challenges encountered in large data processing centres (datacentres) when utilizing server virtualization, which often necessitates working in environments comprising thousands of virtual machines [14]. This gives rise to issues pertaining to the scale of *MAC* address tables. With regard to *VLANs*, the limitation arises from the 12-bit *VLAN ID*, which supports a maximum of 4094 distinct networks [15]. Conversely, the deployment of the Spanning Tree Protocol (*STP*) in data centres to prevent link-layer loops has the consequence of rendering many available links inoperable.

These issues have thus prompted the necessity for the development of a network technology that would facilitate the interconnection of physical servers via an IP network, which in turn requires the utilisation of routing protocols at the layer 3 (*L3*) level. This would obviate the disuse of *STP*-generated links to avoid *L2* loops, and more complex routing strategies (such as equal-cost multi-path routing, or *ECMP*) could be employed to assist in the distribution of the network load across all links. Nevertheless, the utilisation of a *L2* for direct communication between virtual machines would remain a requisite [13].

In light of these considerations, *VXLAN* was developed with the objective of addressing the scalability limitations inherent to networks based on *VLANs*. This is achieved through the use of a 24-bit network identifier (*VNI*), which allows for the creation of over 16 million virtual networks that can coexist within the same administrative domain [3]. Furthermore, *VXLAN* employs *L3* routing, which eradicates link wastage and facilitates a more optimal utilisation of the available resources. Additionally, it eliminates the issue of the size of the *MAC* address tables in the switches, which were required to store the *MAC* addresses of all the server virtual machines that were interconnected by each switch.

VXLAN employs the *MAC-in-UDP* encapsulation technique (Figure 5). The link layer packet, which contains the MAC addresses of the source and destination host, is augmented with the addition of a *VXLAN* header. This set is incorporated into the data field of a User Datagram Protocol (*UDP*) datagram, with *IP* employed as the network layer protocol. Subsequently, the packet transmitted to the *IP* network is identified by source and destination *MAC* addresses corresponding to the (*VXLAN* Tunnel End Points) *VTEPs* situated behind the source and destination hosts, respectively. This enables the initial packet to be routed over an *IP* network, allowing it to reach the *VXLAN* hosts in a manner analogous to if it were on the same *LAN*:

A *VXLAN* scheme is comprised of three fundamental elements: the *VXLAN* header, the *VTEP*, and the *VXLAN* Gateways.

In a *VXLAN* network, each virtual network is identified by a 24-bit *VNI*, which allows for the creation of up to 16 million virtual networks. This makes it an optimal solution for scenarios where a substantial number of isolated networks are necessary, such as in data centers with virtual machines. It enables the coexistence of distinct logical networks on a single physical infrastructure.

The operation of *VXLAN* is contingent upon the utilisation of devices designated as *VTEP*. Such devices are responsible for the encapsulation and subsequent de-encapsulation of Ethernet traffic into *UDP* datagrams, which are then transmitted over the underlying *IP* network. *VTEPs* serve as both entry and exit points for *VXLAN* tunnels, enabling connected devices to communicate over an *IP* network as if they were on the same *LAN* (Figure 6).

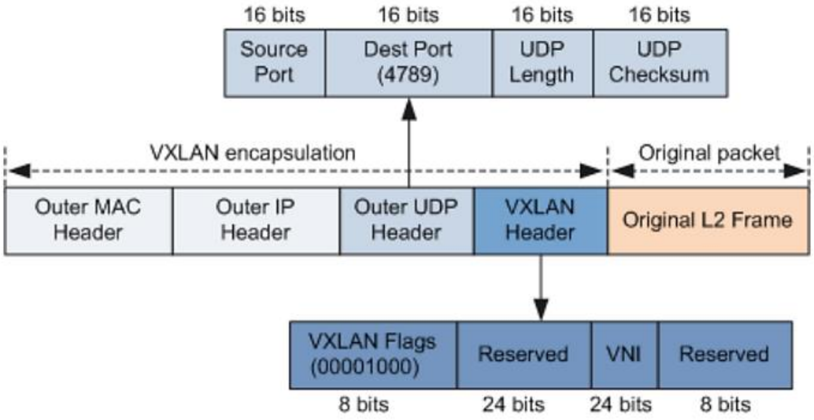

Figure 5. *VXLAN* encapsulation [16]

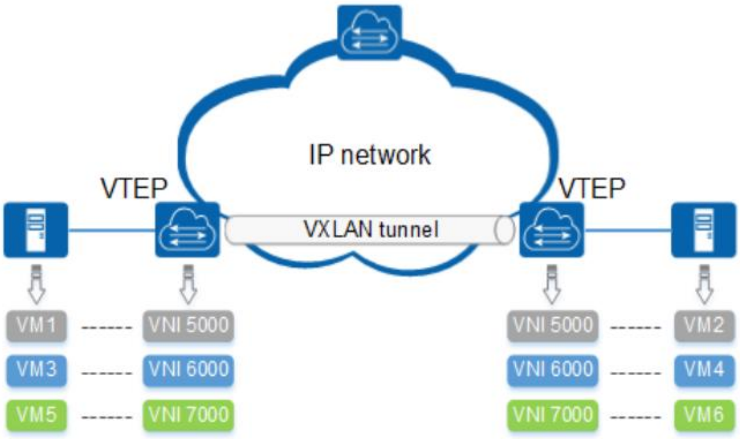

Figure 6. *VXLAN* scheme [16]

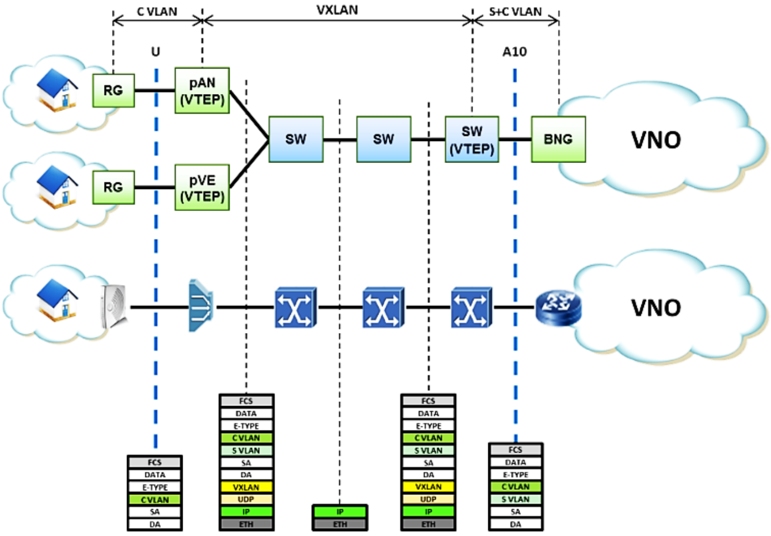

Figure 7. *VXLAN* scheme in *FANS* [3]

3.1 Interconnection of VXLAN model

The *VXLAN* model describes a methodology whereby data is tunnelled from the *OLT* to the service provider. In this process of tunnelling, an identifier is employed to delineate the tunnel, and the aggregator equipment is required to merely accommodate this tunnelling. The scheme is presented in the Figure 7.

As illustrated in the schematic Figure 7, the *C-VLAN* tag information is transmitted through the network. In the downstream direction, the *VXLAN* tag information is incorporated into the Ethernet frame at the adjacent switch, which serves as a *VTEP* at reference point A10. The aforementioned tag information is retained at the *pAN* and functions as a *VTEP*. In contrast, in the upstream direction, the *S-VLAN* and *O-VLAN* tag information is incorporated into the *C-VLAN* tag within the Ethernet frame at the *pAN*. It is crucial to highlight that the *VXLAN* data is discarded on the switch situated in proximity to the A10 reference point, whereas the *S-VLAN* information persists in traversing the *VNO* network.

3.2 Advantages of VXLAN

- Massive scalability: The utilisation of a 24-bit *VNI* enables *VXLAN* to accommodate millions of virtual networks, a capacity that far exceeds the limitations of traditional *VLANs*.
- Efficient utilisation of network resources: In contrast to traditional *VLANs*, *VXLAN* employs *IP* routing, thereby circumventing the constraints imposed by protocols such as *STP* and optimising the utilisation of available links within the physical network.
- Flexibility: *VXLAN*'s use of an IP infrastructure enables the interconnection of geographically dispersed networks, obviating the need to rely on the physical network topology.
- Virtual machine mobility: *VXLAN* simplifies *VM* mobility by allowing *VMs* to retain their *IP* addresses while being moved across physical servers, facilitating load balancing and maintenance without disrupting service [6, 17].

3.3 Limitations and technical challenges of VXLAN

- Configuration complexity: *VXLAN*, being a more sophisticated technology, necessitates a more intricate configuration of the *VTEP* device and a more sophisticated approach to traffic management.
- The additional processing required for encapsulation: The additional encapsulation of Ethernet traffic within *UDP* datagrams can result in network overhead, which may have an adverse impact on performance in certain circumstances.
- IP network support requirements: For the underlying *IP* network to function correctly, it must support multicast routing, which may not be available in all network infrastructures.
- Dependency on underlying infrastructure: *VXLAN* relies on a robust Layer infrastructure. If the underlying network does not support sufficient routing capabilities or lacks proper multicast configurations, it may hinder the effectiveness of *VXLAN* [6, 12].
- Multicast traffic challenges: In environments with significant multicast traffic, *VXLAN* may introduce complications due to the need for ingress replication for broadcast, unknown unicast, and multicast (*BUM*) traffic, potentially leading to inefficiencies [18].

4. MPLS MODEL

The Multiprotocol Label Switching (*MPLS*) model represents a standardised data transport mechanism, originally developed by the Internet Engineering Task Force (*IETF*) and formally defined in *RFC* 3031 [19]. It operates between the data link layer and the network layer of the Open Systems Interconnection (*OSI*) model. The *MPLS* model was developed with the objective of providing a unified system for data transport over networks that may include both circuit and packet traffic. The principal objective is to enhance the efficiency and velocity of routing decisions, which are based on the utilisation of labels that are assigned to data packets. It is capable of carrying a variety of traffic types, including voice and *IP*. *MPLS* was the logical successor to Frame Relay and *ATM*, becoming the preferred technology for high-speed data and digital voice on a single connection. It offers enhanced reliability and performance, and can potentially reduce transport costs due to its increased network efficiency and prioritisation capabilities.

The deployment of *MPLS* in *IP* networks offers a number of advantages in the context of *VPN* (Virtual Private Network) creation, traffic engineering, fault protection mechanisms, *QoS* support, multi-protocol support, and Class of Service (*CoS*) establishment. Its basic features include integration of *L2* (data link) and *L3* (network) of the *OSI* model, with optimisation of routing achieved through reduction in algorithmic complexity and maintenance of communication state between two nodes [20].

MPLS is a scalable, protocol-independent technology. In an *MPLS* network, data packets are assigned labels, which serve to identify and categorise them. The determination of packet forwarding is based exclusively on the information contained within the label, obviating the necessity for an examination of the packet itself. This enables the creation of end-to-end circuits over any transport medium and the utilisation of any protocol. The principal advantage is the elimination of reliance on a specific OSI model data link *L2* technology, such as *ATM*, Frame Relay, *SONET* or Ethernet, and the avoidance of the necessity for multiple *L2* networks to cater to disparate types of traffic. *MPLS* is a member of the packet-switched family of networks.

MPLS operates at a layer that is generally considered to fall between the traditional *OSI L2* and *L3* protocol. The objective of the design was to provide a unified data transport service for both circuit-based clients and packet-switched clients, offering a datagram service model. It is capable of carrying a multitude of different types of traffic in a native state.

The similarity of *MPLS* with *ATM* and Frame Relay lies in the fact that, at each hop across the network, the label value in the header is changed, in contrast to the forwarding of *IP* packets. The evolution of *MPLS* technologies has been informed by a consideration of the relative strengths and weaknesses of *ATM* which has resulted in *MPLS* becoming the dominant technology in this field. The objective has been to develop a solution with a lower overhead than *ATM*, while providing connection-oriented services for variable-length frames. In particular, *MPLS* eliminates the signalling and switching protocol that is characteristic of *ATM*. *MPLS* acknowledges that the use of small *ATM* cells is unnecessary in the core of modern networks, given the high speeds and lack of queuing delays observed in modern optical networks. This is in contrast to the motivation behind the cellular nature of *ATM*, which was to reduce delays to support voice traffic [21]. Concurrently, *MPLS* strives to maintain the traffic engineering and out-of-band control that initially made *ATM* and Frame Relay appealing for implementation in large-scale networks as can be seen in the Figure 8.

As can be seen in the Figure 9, *MPLS* operates by inserting prefixes into packets with a header of their own, containing one or more labels. These are collectively referred to as a label stack, which comprises four fields for each entry.

- A 20-bit label value is employed, with value 1 representing the router alert.
- A three-bit field is also included, which is used for the purposes of differentiating between different traffic classes in order to facilitate the implementation of *QoS* policies and explicit congestion notification (*ECN*).
- A one-bit indicator at the bottom of the label stack indicates that the current label is the last in the sequence.
- An 8-bit time-to-live (*TTL*) field is also included.

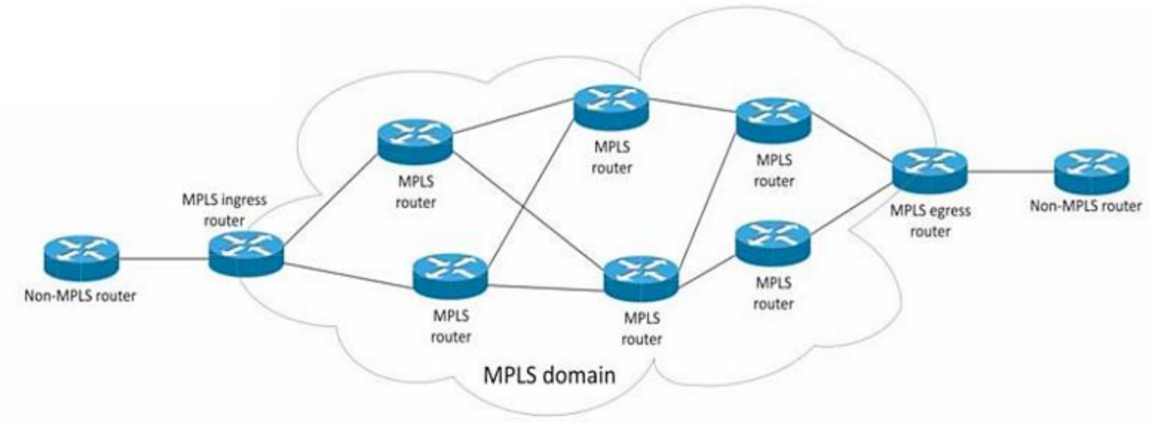

Figure 8. *MPLS* scheme [22]

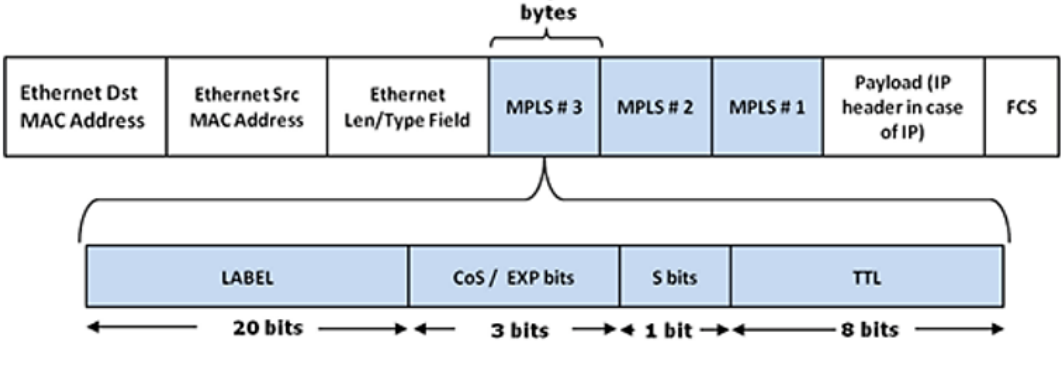

Figure 9. *MPLS* encapsulation [23]

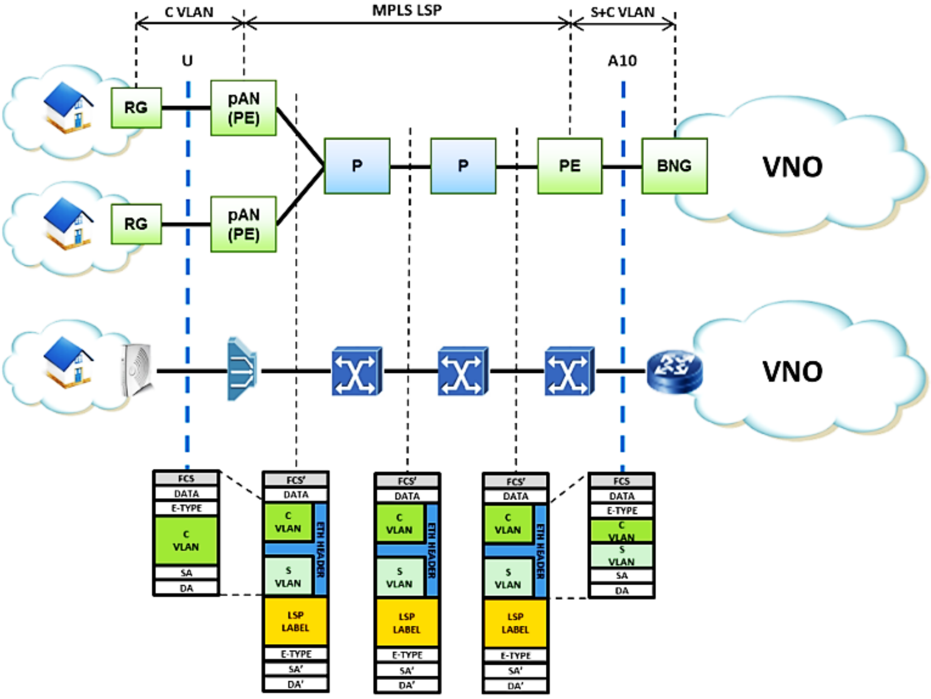

Figure 10. *MPLS* scheme in *FANS* [3]

4.1 Interconnection of MPLS model

Another potential technique for managing operator data flows in a *FANS* scenario is a label-based switching approach, such as *MPLS*, as illustrated in the Figure 10.

The principal distinction between the MPLS scheme and the *VLAN* model is the existence of an *MPLS* tunnel (defined by *LSP* labels) that encompasses both the *C-VLAN* label and the *S-VLAN* label information.

It can be observed that the *C-VLAN* label information is transmitted throughout the network. In the downstream direction, the *MPLS LSP* information is incorporated into the Ethernet frame at the *PE* router situated in proximity to the A10 reference point. The aforementioned information labels are conveyed up to the point at which the *pAN* assumes the role of the *PE* router. In contrast, in the upload direction, the *S-VLAN* information is incorporated into the *C-VLAN* label within the Ethernet frame at the *vAN*, while the *MPLS LSP* information is incorporated into the Ethernet frame at the *pAN*. It is crucial to highlight that the *MPLS* label information is discarded at the switch situated adjacent to the A10 reference point, whereas the *S-VLAN* information persists in traversing the *VNO* network. Nevertheless, despite the flexibility and scalability of the MPLS network architecture, in a purely *L2* evolution context, it may be advantageous to consider utilising the *O-VLAN* scheme instead of the L2.5 *MPLS* extension to the access network. Another motivation for this approach is that current access nodes may not support *MPLS*. However, the addition of *MPLS* capability may result in increased complexity and cost for the node. The above description is shown in the Figure 10.

4.2 Advantages of MPLS

- Efficient transport of data: It is facilitated by the utilisation of *MPLS* which is capable of efficiently handling a variety of traffic types, including voice, data, *IP*, and others, through the use of labels that facilitate the packet forwarding process.
- Multi-protocol support: *MPLS* is able to function effectively over a wide variety of underlying protocols, thereby conferring considerable flexibility with regard to the types of networks in which it can be deployed.
- Optimisation of traffic flows: The capacity to implement traffic engineering policies and support *CoS* renders it a favoured option for networks that necessitate traffic prioritisation and a superior quality of service.
- Reliability and security: MPLS guarantees packet delivery through its label-based routing mechanism. This reliability is backed by *SLAs* (Service Level Agreement) that ensure the provider resolves outages promptly, enhancing customer trust [6, 17].

4.3 Limitations and technical challenges of MPLS

- Complexity of deployment: It is a significant factor to be considered. The deployment and management of *MPLS* is a more complex process than that of other technologies, such as *Q-in-Q* or *VXLAN*. This is due to the necessity for a more sophisticated infrastructure and the requirement for greater configuration and management efforts.
- Hardware requirements: In order to support *MPLS*, routers and switches must be *MPLS*-compatible, which can require costly hardware upgrades in older networks.
- Administrative overhead: The management of *MPLS* labels and routes can be complex, particularly in large-scale networks where a high volume of traffic is handled.
- Limited bandwidth options: Upgrading bandwidth in an *MPLS* network can be costly and may not always be feasible due to infrastructure constraints. This limitation can restrict growth potential for businesses relying on high-speed connectivity [6, 17].

5. COMPARISON BETWEEN THE THREE MODELS: Q-IN-Q, VXLAN AND MPLS

The three *FANS* models (*Q-in-Q*, *VXLAN* and *MPLS*) present distinct approaches to the shared network management and logical partitioning of resources in a fiber optic infrastructure. Each model has characteristics that render it suitable for specific scenarios, offering a range of advantages and challenges.

5.1 Scalability

In terms of scalability, the *VXLAN* model offers the highest capacity, allowing the creation of up to 16 million virtual networks. This is made possible by the use of a 24-bit network identifier, which is capable of accommodating a significantly larger number of networks than other models. This figure is considerably greater than the 4096 *VLAN* limitation of the *Q-in-Q* model, even when employing multiple labels. While *MPLS* does not facilitate the management of virtual networks in the same manner, it does provide scalability through the efficient transportation of a multitude of traffic types on a unified infrastructure. Accordingly, *VXLAN* is the optimal selection for scenarios where the number of virtual networks is a pivotal consideration, such as in expansive data centers.

5.2 Implementation complexity

MPLS is the more complex model in terms of implementation, as it necessitates the presence of specific infrastructure and the configuration of numerous parameters pertaining to forwarding labels and traffic engineering. Although *Q-in-Q* and *VXLAN* also present configuration challenges, both models are based on more widely used standards and are therefore more readily integrated into networks that already utilise Ethernet or *IP* networks. *Q-in-Q* is particularly straightforward to implement in Metro Ethernet networks, whereas *VXLAN* benefits from the use of *IP* routing, which offers greater flexibility in terms of the physical topology of the network.

5.3 Isolation and security

With regard to the issue of traffic isolation, the three models in question achieve this objective in disparate ways.

The *Q-in-Q* method employs the use of multiple *VLAN* tags to segregate the data traffic of disparate operators, thereby offering a robust level of isolation in scenarios where the underlying physical network is shared. In contrast, *VXLAN* offers enhanced isolation due to its overlay methodology and the utilisation of *VNI* for the identification of virtual networks. The use of labels and the capacity to accommodate *VPNs* make *MPLS* an effective solution for networks where security and quality of service are paramount.

Summarising the relevant aspects for the analysis of this point, it can be seen:

- **MPLS** provides inherent security through its ability to create isolated paths for different customers or services.
- **VXLAN** offers enhanced isolation by allowing multiple tenants to coexist on the same physical infrastructure without interfering with each other's traffic.
- **Q-in-Q** ensures that even if customer *VLAN IDs* overlap, their traffic remains segregated through the use of outer tags.

5.4 Complexity and costs considerations

The implementations carried out in the IPlan company [6] in terms of complexity and costs allowed the construction of Table 1.

Table 1. Complexity and costs resulting from the implementation carried out

Features	Q-in-Q	MPLS	VXLAN
Complexity of Implementation	Low to moderate	High	Moderate to high
Configuration Complexity	Relatively simple, requires VLAN tagging	Complex, requires detailed configuration for routing and traffic engineering	Moderate, requires knowledge of encapsulation and network design
Operational Complexity	Low, mainly involves VLAN management	High, ongoing management of SLAs, routing protocols, and performance monitoring	Moderate, needs management of virtual networks and overlays
Cost of Maintenance	Low, straightforward maintenance	High, ongoing costs for maintenance contracts and equipment upgrades	Moderate, depends on the scale and complexity of the deployment
Scalability	High (up to 16 million VLANs)	Moderate, limited by label space	Very high (up to 16 million VNIs)
Use Case Suitability	The best for service providers needing VLAN extension without complex routing requirements	Ideal for enterprises needing reliable, high-performance WAN connectivity with QoS	Suited for cloud environments requiring flexible, scalable layer 2 networks

5.5 Technical comparison

The comparative Table 2 is shown below as a summary comparison.

5.6 Use cases

Each model has specific use cases where it is particularly effective.

Q-in-Q: Ideal for Metro Ethernet networks, where a simple and efficient way to manage multiple *VLANs* from different carriers in a shared infrastructure is required. In contexts where Ethernet compatibility is of paramount importance and scalability is not a primary consideration, this model is the optimal choice.

VXLAN: In scenarios that necessitate high scalability and flexibility, it is the optimal model. It is particularly beneficial in data centres and in networks where virtual machines are utilised, necessitating an overlay network to interconnect servers across *IP* infrastructures.

MPLS: Better suited to networks that require the handling of diverse traffic types with varying priorities. The capacity to implement traffic engineering and *QoS* policies renders it an optimal choice for networks where the accurate management of voice, video and data traffic is paramount.

Table 2. Comparison summary

Features	Q-in-Q	MPLS	VXLAN
Primary Use Case	Service provider VLAN extension	Enterprise WAN connectivity	Cloud data center virtualization
Scalability	Up to thousands of VLANs	Limited by label space	Up to 16 million VNIs
Traffic Isolation	Yes	Yes	Yes
QoS Support	Limited	Extensive	Limited
Complexity of Implementation	High	Low	Moderate
Cost Consideration	Generally low	High, due to specialized hardware	High, due to complexity and requirements

6. CONCLUSIONS

A comparative analysis of the three interconnection models of the *FANS* standard reveals that each possesses distinctive strengths, rendering them suitable for disparate network sharing scenarios. *Q-in-Q* represents an efficient and straightforward solution for Metro Ethernet networks, offering an appropriate degree of isolation and scalability within existing Ethernet infrastructures. In contrast, *VXLAN* is the optimal selection for scenarios that necessitate a substantial number of virtual networks, such as data centers and networks supporting virtual machines. This is due to its capacity to construct an overlay on top of *IP* infrastructures. Finally, *MPLS* is the optimal solution for networks where quality of service and traffic prioritisation are paramount, offering enhanced flexibility in managing diverse traffic types.

Furthermore, the implementation of these models in shared networks offers not only technological improvements but also economic benefits. The sharing of physical infrastructure allows operators to reduce the costs associated with the construction and maintenance of separate networks. Additionally, *FANS* enables the development of novel customer services, thereby fostering more dynamic competition at the active layer of networks. In conclusion, the selection of a model is contingent upon the specific requirements of the network environment and the priorities of operators with regard to scalability, cost and complexity of deployment. This emphasises the necessity of adapting the infrastructure to the particular needs of each operator.

REFERENCES

[1] FCGA. (2015). FTTH Council Global Alliance – FCGA, FTTH Council - Definition of terms. https://data.nag.wiki/AlphaMile/Certificates/FTTH_Definition_of_Terms-Revision_2015-Final.pdf.

[2] Fiber Optic Association. (2016). Fiber optic networks. https://www.thefoa.org/Lennie/nets.html.

[3] Broadband Forum. (2020). Fixed access network sharing - Architecture and nodal requirements (FANS). Technical Report TR-370i2. https://www.broadband-forum.org/pdfs/tr-370-2-0-0.pdf.

[4] ITU-T. (2008). Gigabit-capable passive optical networks (GPON): General characteristics. Recommendation ITU-T G.984.1. https://www.itu.int/rec/T-REC-G.984.1-200803-I/en.

[5] ITU-R. (2015). IMT vision – Framework and overall objectives of the future development of IMT for 2020 and beyond. ITU-R Recommendation M.2083-0. Electronic Publication, Geneva. https://www.itu.int/dms_pubrec/itu-r/rec/m/r-rec-m.2083-0-201509-i!!pdf-e.pdf.

[6] IPLAN (Internet Service Provider – Large Enterprise Data Center). https://www.iplan.com.ar/Acerca-de-iplan.

[7] IEEE Standards Association. (2006). IEEE 802.1ad-2005. https://standards.ieee.org/standard/802_1Q-2011.html.

[8] Juniper Networks. (2024). Configuring Q-in-Q tunneling and VLAN Q-in-Q tunneling and VLAN translation. In Ethernet Switching User Guide. Juniper Networks, Inc., Sunnyvale, California, p. 910. https://www.juniper.net/documentation/us/en/software/junos/multicast-l2/topics/topic-map/q-in-q.html.

[9] Cisco. (2022). Configuring Q-in-Q and layer 2 protocol tunneling. In Cisco Catalyst IR8340 Rugged Series Router Software Configuration Guide, Cisco IOS XE Release 17.8.x. Cisco Systems, Inc., San Jose, CA, p. 161. https://www.cisco.com/c/en/us/td/docs/routers/ir8340/software/configuration/b_ir8340_cg_17-8/m-configuring-q-in-q-and-layer-2-protocol-tunneling.html.

[10] Netberg. Q-in-Q configuration tips. https://netbergtw.com/top-support/articles/q-in-q-configuration-tips/.

[11] Huawei Enterprise. (2021). Overview of QinQ. https://support.huawei.com/enterprise/en/doc/EDOC1100137875/ded44dec/overview-of-qinq.

[12] Juniper Networks. (2024). Understanding VXLANs. https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/topic-map/sdn-vxlan.html.

[13] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., Wright, C. (2014). Virtual eXtensible Local Area Network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3 networks. RFC 7348. https://doi.org/10.17487/rfc7348

[14] Arista Networks. Data Center Interconnection with VXLAN. http://allvpc.net/Arista_Design_Guide_DCI_with_VXLAN.pdf.

[15] Juniper Networks. (2020). What is VXLAN? https://www.juniper.net/us/en/research-topics/what-is-vxlan.html.

[16] Huawei. (2024). What is overlay network? https://info.support.huawei.com/info-finder/encyclopedia/en/Overlay%20network.html#content1.

[17] Sptel – Reliability with an Edge. (2021). What is MPLS (MultiProtocol Label Switching) for business? https://sptel.com/mpls-advantages-and-disadvantages/.

[18] Dube, C. (2021). Project 01. VXLAN use case study. University of New Hampshire, Connectivity Research Center. https://connectivity.unh.edu/research-projects/next-generation-data-communication-protocols/vxlan-use-case-study.html.

[19] Rosen, E.C., Viswanathan, A., Callon, R. (2001). Multiprotocol label switching architecture. RFC Editor: RFC 3031. https://doi.org/10.17487/RFC3031

[20] De Ghein, L. (2016). MPLS Fundamentals. Cisco Press, Hoboken, NJ.

[21] Goldman, J.E., Rawles, P.T. (2004). Applied Data Communications: A Business-Oriented Approach. Hoboken, NJ: Wiley.

[22] Hunter, P. (2024). MPLS — What is it? In plain English, please…. In SCTE TechExpo 24, Atlanta, Georgia. https://broadbandlibrary.com/mpls-what-is-it-in-plain-english-please/.

[23] GL Communications Inc. (2024). Multi-protocol label switching (MPLS) testing features. https://www.gl.com/telecom-test-solutions/ethernet-networks-devices-switches/multi-protocol-label-switching-mpls-packetexpert.html.

NOMENCLATURE

5G	Fifth Generation of Wireless Cellular Technology
ATM	Asynchronous Transfer Mode
BUM	Broadcast Unknown Unicast and Multicast
C-VLANs	Client VLANs
CoS	Class of Service
ECMP	Equal-Cost Multi-Path
ECN	Explicit Congestion Notification
FANS	Fixed Access Network Sharing
FTTH	Fiber to the Home
GPON	Gigabit Passive Optical Networks
ID	Identification
IDs	Identifications
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
InP	Infrastructure Provider
InPs	Infrastructure Providers
IoT	Internet of Things
IP	Internet Protocol
ISPs	Internet Service Providers
ITU-R	International Telecommunication Union - Radiocommunication Sector
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
L2	Layer 2
L3	Layer 3
LAN	Local Area Network
LLC	Logical Link Control
LSP	Label Switched Path
MAC	Media Access Control
MAC-in-UDP	Media Access Control in User Datagram Protocol
mMTC	Massive Machine-Type Communication
MPLS	Multiprotocol Label Switching
NaaS	Network as a Service
O-VLAN	Operator VLAN
OLT	Optical Line Termination
ONT	Optical Network Terminal
ONTs	Optical Network Terminals
OSI	Open Systems Interconnection
pAN	physical Access Node
PE	Provider Edge Routers
PON	Passive Optical Networks
Q-in-Q	802.1Q tunnelling
QoS	Quality of Service
RFC	Request for Comments
S-VLANs	Service VLANs
SAMAC	Source MAC Address
SLAs	Service Level Agreement
SONET	Synchronous Optical Network
STP	Spanning Tree Protocol
TR-370	Fixed Access Network Sharing - Architecture and Nodal Requirements
TTL	Time To Live
UDP	User Datagram Protocol
vAN	virtual Access Node
VLAN	Virtual Local Area Network
VLANs	Virtual Local Area Networks
VM	Virtual Machine
VMs	Virtual Machines
VMware	Virtual Machine and software
VNI	VXLAN Network Identifier
VNO	Virtual Network Operator
VNOs	Virtual Network Operators
VPN	Virtual Private Network
VTEP	VXLAN Tunnel End Point
VTEPs	VXLAN Tunnel End Points
VXLAN	Virtual Extensible Local Area Network