

# SVPLS: Software VPLS

Luis Miguel Diaz, Francisco Valera, Arturo Azcorra

**Abstract**— This article presents a development to provide a virtual private LAN service (VPLS) over IP, together with the trials carried out within the framework of the 6th FP European Network of Excellence, E-NEXT. Shortly after the experiences performed with different IETF drafts in commercial equipments installed over the optical metropolitan network deployed in project PREAMBULO (national research project granted by the Spanish Science and Technology Ministry) it was clearly seen the necessity of a more flexible development easily adaptable to the fast changing solutions appearing in IETF (mainly in l2vpn charter) and capable of testing these new proposals without the requirement of expensive commercial equipment which is not always available and which is not always provided with all the up-to-date draft alternatives. This article shows the technical options adopted for SVPLS (Software VPLS) as well as the different tests that have been made to validate the service.

**Index Terms**— L2VPN, VPLS, Ethernet, broadband,

## I. INTRODUCTION

Since the appearance of the first corporate networks in the 80's, and as far as the enterprises began to expand though different sites in different locations there has always exist the demand for a technology to allow an easy way to connect all their local networks as if they were unified into a single private network (a virtual private network or VPN).

There are many different ways to provide such a VPN service, depending on the protocol service that is wanted to be provided (layer 2 or layer 3), depending on who is assuming the responsibility of the service provisioning (user provisioned or provider provisioned), depending on how the provider network is developed (circuit based or packet based), etc. Section 2 in this article includes a short review of VPNs technology mainly focusing on layer 2 solutions.

Following sections are focusing on one of the alternatives that is becoming more and more relevant from the providers point of view as long as their networks are being migrated to IP or IP/MPLS technology and as long as Ethernet is

gaining more and more presence in the access networks: the Virtual Private LAN Service (VPLS). It describes a multipoint to multipoint Ethernet scenario where the provider network would just be seen as a switch that allows the clients to connect their Ethernet LANs altogether transparently.

After the different trials performed over PREAMBULO [1] metropolitan network, Fig. 1, with commercial solutions (provided by Nortel and Timetra), based on current IETF draft document and (see [2] for further details on these tests) it was clearly seen that in order to be able to keep on researching on VPLS technology (it is not a standard solution and there are different proposals and problems that must still be considered) it was required to develop the solution flexible enough so that it could be easily adapted to any possible incoming change, and so that it to could be used to test any particular commercial implementation. This article is describing the first prototype developed towards this direction, allowing a Linux PC platform to perform as a provider edge VPLS equipment.

In section 3, this solution is explained, beginning with the definition of the service that is going to be provided and then with a comparison with standard VPLS. After a detailed description of the software implementation, the section ends with the different trials designed and performed so as to validate the proposal within the framework of E-NEXT Network of Excellence [3].

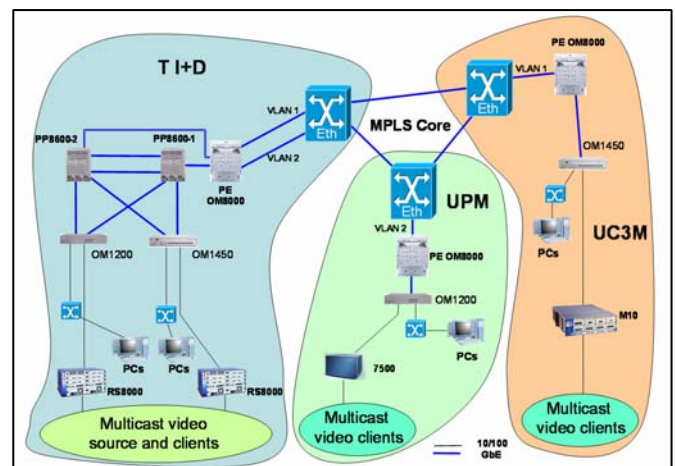


Fig. 1. VPLS testbed trialed in PREAMBULO DWDM metropolitan infrastructure [2]

L.M. Díaz is with Telefónica I+D, Emilio Vargas 6, 28048 Madrid, Spain; (e-mail: luismdv@tid.es).

F. Valera and A. Azcorra are with the Universidad Carlos III of Madrid, Avda. de la Universidad 30, 28911 Leganés, (Madrid) Spain (e-mail: fvalera;azcorra@it.uc3m.es).

This article has been partially granted by the Spanish Science and Technology Ministry research and development program (period 2000-2003) through project PREAMBULO from where it was possible to get the original ideas to develop SVPLS. Most of the tests were possible thanks to the collaboration of Telefónica I+D through an agreement within the framework of E-NEXT network of excellence, funded by the European Union IST program. Authors F.Valera and A. Azcorra want to thank also project CAPITAL (MEC, TEC2004-05622-C04-03/TCM) for the partial financial support.

Finally the article includes the most important conclusions achieved during the realization of this development.

## II. LAYER 2 VIRTUAL PRIVATE NETWORK

### A. Introduction

The idea of creating VPNs over a public shared provider infrastructure may be developed over diverse solutions and all of them rely on the same ‘tunnel’ concept: different machines located in the border of certain entities (tunnel servers) establish normal connections but instead of carrying typical data in the payload field of the protocol used in their connections, what they are really carrying are client packets that belong to other communications. From the point of view of the clients, the tunnel is absolutely transparent and communication between clients located in remote sites happens as if they were located in the same network.

The initial approaches to these tunnelling mechanisms and to VPNs were based on dedicated circuits provided by the operator (typically TDM leased private lines, Frame Relay service or ATM).

Although these alternatives were effectively deployed and allowed to share the provider infrastructure in order to deliver a packet service, the number of problems they are showing nowadays (high costs, bandwidth inflexibility, provider migration to IP/MPLS, etc.) forced the movement towards other solutions.

The most immediate solution was the usage of user provisioned VPNs based on tunnelling mechanisms such as PPTP, ATMP, L2TP or IPsec. They were able to provide either Layer 2 or Layer 3 VPNs in small environments, but as far as the number of peer networks begins to grow the number of tunnels required to keep all of them connected becomes unmanageable for the client which was assuming the responsibility of VPN provisioning.

As providers were able to migrate their circuit-based transport networks to IP or MPLS/IP networks, it was feasible for them to recover the provision of VPNs and different technical solutions appeared so as to support both layer 2 services (L2VPN) and layer 3 services (L3VPN). Although both technologies have advantages and disadvantages and in fact their objectives are different, L2VPN is acquiring more and more interest since important effort is been placed on the installation of a LAN technology such as Ethernet into the access network and so more and more Ethernet services are being demanded (from point-to-point to multi-point-to-multipoint).

The IETF has been proposing certain solutions and although the point-to-point scenario is already solved based on the well known Martini tunnelling mechanism [4], the extension to multi-point to multi-point environments, VPLS, is not so clear yet.

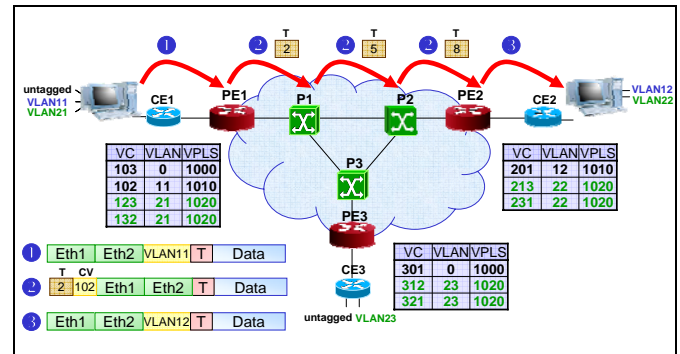


Fig. 2. VPLS example

The basic Virtual Private LAN Service, described in IETF draft [5], is also known as Transparent LAN service, and specifies the provisioning of a LAN service to customers that may be spread across a metro or wide area network. As the classical VLAN technology it allows several LAN to work as a single LAN, but instead of a layer 2 connection between the different LANs based on VLAN aware switches, this service allows the different LAN segments to be connected through a packet-switched network (see Fig. 1 for example where the different participants were sharing the same LAN through an IP/MPLS core network). And moreover, it would allow the connection of different VLAN segments across the network (transparently carrying the VLAN frame or just the Ethernet frame, depending on the operation mode).

Fig. 2 is showing an example of the transmission of an Ethernet frame between two terminal equipments.

Previous messages, not shown in the figure, would have signalled the relation between every virtual circuit (VC) and its corresponding VPLS instance identifier using different protocols, depending on the VPLS specification used (draft [5] proposes BGP while draft [6] proposes LDP). PE1 for example would have signalled the pairs [102,1010] and [123,1020] to PE2 and [103,1000] and [132,1020] to PE3).

In the example, the PC located behind CE1 is sending an Ethernet frame towards a PC located behind CE2. Since PC1 is marked with VLAN-ID 11, PE1 is able to identify it as belonging to VPLS instance number 1010 and to tag it with its corresponding virtual circuit number 102 (it strips the VLAN-ID before sending it to the core network).

The frame would then go through the network with the typical MPLS label swapping mechanism and once in the PE2 the virtual circuit number 102 would identify the VPLS instance and PE2 would then know the destination belongs to VLAN number 12.

VPLS drafts have been accepted by the L2VPN IETF charter but there are still pending issues that have to be solved (label signalling protocol, auto-discovery mechanism, scalability problems, etc.).

Commercial equipments have already started to implement VPLS, some of them following the drafts and some other following proprietary solutions. However these solutions are not usually interoperable and since they are only implemented on commercial equipment, it is very difficult and expensive to perform research and progress on other alternatives.

The following section is describing the software solution that has been implemented assuming certain restrictions and modifications over the standard VPLS that will be commented.

### *B. Service definition*

In order to define SVPLS service, three main aspects were taken into account:

- 1) This SVPLS service must work as any other VPLS service. From customer view, no difference should be advisable.
- 2) As it is based on a PC platform, simplifications will be needed in order to make the service work.
- 3) Since VPLS drafts do not set MPLS as mandatory protocol, the core network will be IP based (the specification is just mentioning 'a packed switched network').

Since SVPLS must work as VPLS, MAC learning and traffic replication must be present. Every customer MAC must be learned and assigned to the PE that serves these customers and as it is IP based, every PE will be identified by its own IP address, so every MAC must be assigned to one of these IP addresses.

Because a PC only has a few slots to add network cards, VLAN support is mandatory in SVPLS in order to support many customers per PE. One network interface can be divided in many logical interfaces, and several different customers (VPNs) can be attached to the same physical port. This way, interface and VLAN tag identify one specific VPN (one interface can support up to 4094 different VPNs).

Regarding forwarding, signalling and auto-discovery, several changes were applied.

First of all, a forwarding mechanism must be defined, because the core network is based on IP and the double MPLS tag cannot be used. A different tunnelling mechanism must be used to forward traffic across this core, and since PEs are identified by their IP addresses, and these IPs are known by the rest of PEs, an IP based tunnel is the easiest way to connect them. The core network will only see traffic crossing the network from one edge to another. Instead of IP-IP tunnels, UDP-IP tunnels were chosen due to their programming advantages. They provide automatic packet fragmentation, which is very useful in this case since extra labels are going to be added in the lower layers and in case data is the same length as in a normal environment the final frame length will probably be no compliant with common switches (unless jumbo frames are enabled within the whole network). The overhead of 20 bytes imposed by UDP is not significant compared with the advantages that are obtained from this protocol.

The internal MPLS tag is preserved as packet marking mechanism (VPN identification), so outgoing PE can know the VPN each packet belongs to. Any other header could have been added to identify packets, but MPLS was preserved in order to make the convergence from SVPLS to a standard VPLS based on MPLS much easier. The UDP tunnel combined with the internal MPLS tag make the same role as the double MPLS tag, and forwarding/VPN identification can be performed in SVPLS.

Although UDP tunnels require no signalling (the IP

addresses of other PEs are known by configuration or auto-discovery), internal tags may demand some signalling (every PE must be aware of the VPN-ID/tag association). As in every PE the available VPNs must be manually configured (VPN-ID associated to every customer), and every VPN-ID is unique in the whole network, there would be no need to signal them if VPN-ID is also used as MPLS tag, since every PE will use the same tag for identifying one specific VPN. This may cause a scalability problem but 232 available VPNs should be enough for a PC-based service. The only problem of this no-signalling mechanism is that a PE cannot know if there are other PEs with customers in the same VPNs that it has, so it would have to replicate traffic to any other PE in the network. This resource waste is not acceptable, and something must be added to SVPLS to solve the problem: an auto-discovery mechanism.

The auto-discovery is a mechanism used by new PEs to introduce themselves to the network. The new PE sends 'hello' packets to all PEs in the core network so they can add the new PE into their tables (only the new PE requires to have the IP addresses of the other PEs preconfigured). This 'hello' message can also be used to transport extra information so that the new PE can inform about what VPNs it has customers for. As soon as the rest of PEs reply this message with a 'hello-reply', including their own information, the new PE can also know the VPNs in every other PE of the network.

These 'hello' packets can also be used as a keep-alive mechanism if they are periodically sent between PEs. VPN information is then kept up to date (and so, adding new customers to an existing PE becomes an easier task) and if one PE goes down it can be detected by other PEs (since 'hello' packets would not be received within a certain time period).

So finally, the SVPLS service can be summarized as follows: it is based on UDP tunnels over an IP network with internal MPLS tags to identify VPNs. Although no signalling is defined, the auto-discovery and keep-alive mechanism combination are able to maintain updated VPN information and to support dynamic customers, VPNs and/or PEs changes.

### *C. Validation*

In order to validate the service, a testing scenario was created, with three PCs running SVPLS (used as three PEs), interconnected across an IP core, and a traffic generator (Spirent Smartbits 900) used to simulate lots of clients/VPNs.

In addition to conformance tests, it was very important to demonstrate that the software was able to manage a reasonable amount of traffic so the three PEs running the service were chosen to be resource limited (Pentium III, 800MHz, with 128RAM MB).

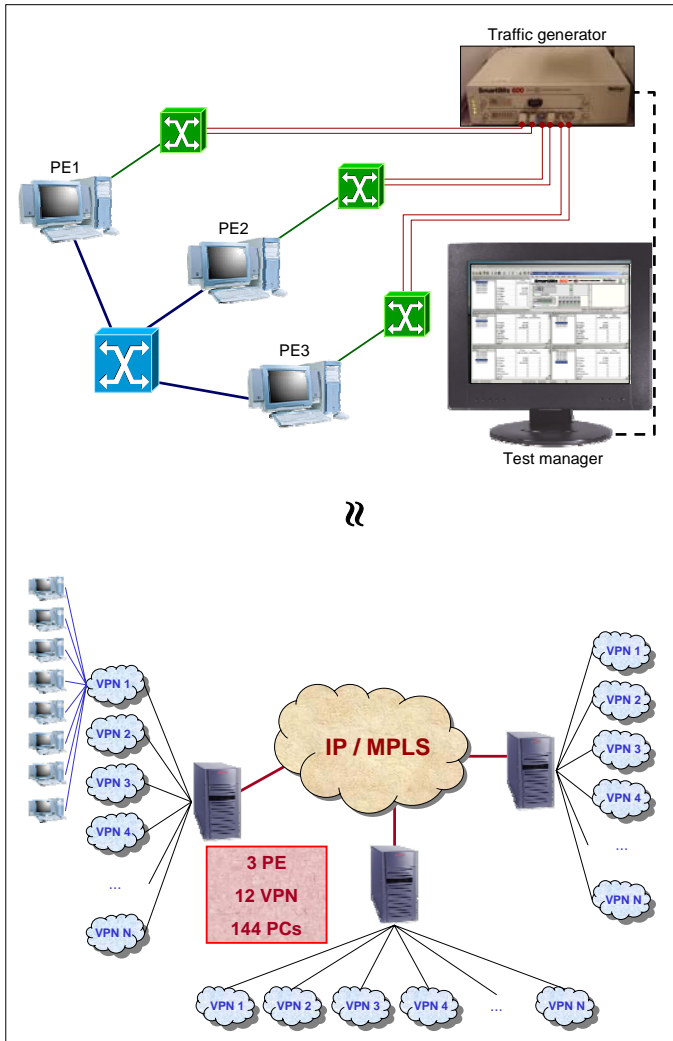


Fig. 3. Test bed and equivalent simulated network

Fig. 3 shows the testbed that was used in the trials. The above picture shows the connections between the different elements: the three PCs running the service, different VLAN aware switches, the traffic generator and another PC in order to control the traffic generator functionality.

This connection schema allowed generating two independent set of flows from every PE towards the two other PEs. Changing the MAC addresses of these flows it is also possible to simulate different user terminals (the traffic generator is able to sequentially change the MAC address according to an established pattern). Fig. 3 is also showing the equivalent simulated network.

Over this testing environment different trials were performed changing traffic injection parameters: packet size, number of VPLS instances, number of VLANs, number of user terminals, throughput.

The most important results from these tests are the following:

- Basic connectivity was tested. Every VPN appear to be connected to the same LAN, over a standard switch. Replication between sites is performed until MAC address is learned. No traffic between different VPNs appears.
- Due to hardware limitations (PC processing capabilities), maximum bandwidth available per

interface (with two interfaces per PE) is 70 Mbps full duplex. This test was performed without SVPLS service in order to acquire a valid reference, and was tested with a constant packet size of 1500 bytes.

- When SVPLS service is started, this maximum bandwidth is reduced (because processing is higher) down to 60 Mbps (best case, without using VLANs) or 50 Mbps (worst case, using VLAN tags). As VLAN support increases processing requirements, performance goes slightly down. This limitation can be easily solved with better PCs (we tested the worst case with low performance PCs).
- Testing a general case, with random packet size (more packets per second with the same bandwidth), performance goes down to 50Mbps (no VLANs) or 40Mbps (with VLANs). The reason is the same as before: more packets per second increases processing requirements, and so, more powerful PCs are required.
- The number of MAC addresses (clients) or VPNs has no impact on SVPLS service performance.

In the following figure, the four graphics summarise the different tests comparing injected traffic versus received traffic. The best situation has only one present VPN and the packet size is high (less packets per second processed). In this case, 180 Mbps full duplex are measured (adding the three PEs traffic) so 60 Mbps full duplex per interface are available. In case VLAN processing is added or packet size is not constant (more packets per second), as processing increases, performance decreases to 50Mbps full duplex per interface. The general case (VLAN and random packet size) leads to 40Mbps full duplex per interface.

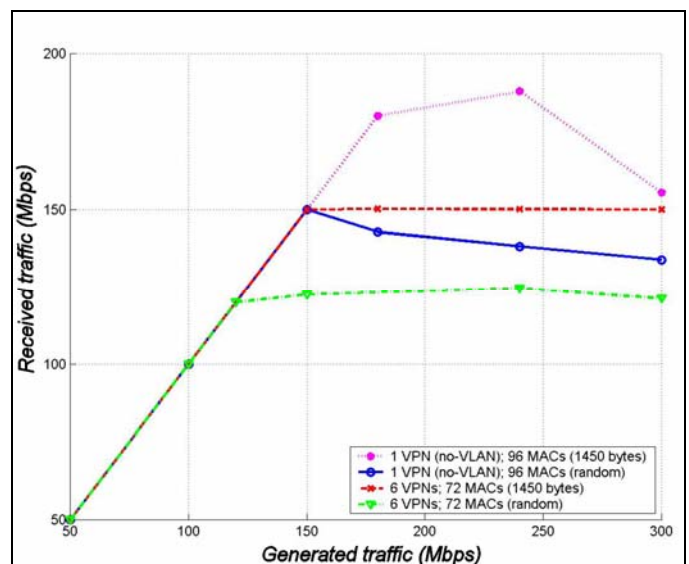


Fig. 4. Forwarded traffic comparison

### III. CONCLUSIONS

SVPLS has been designed to work as a PC based VPLS service over an IP core. Simplifications made to standard VPLS have minimal impact on the service, since customers perceive no difference between SVPLS and VPLS, and the provider is able to offer this service with little effort, because although signalling is reduced (almost erased), the

auto-discovery plus keep alive mechanisms can offer the same results as a VPLS router. The only pre-configuration required by a new PE is the IP addresses of other PEs and some local configuration (exactly the same information a VPLS router needs in order to work). The following list summarises the differences between VPLS and SVPLS:

- 1) MAC learning is performed in SVPLS as in VPLS. This way, traffic is forwarded between PEs (between ports of this 'virtual switch') according to the destination MAC, or flooded (only to those PEs that also have customers for that VPN).
- 2) VLAN support in customer side works the same in SVPLS than in VPLS.
- 3) External MPLS tagging is not used. UDP tunnels are used instead so as to forward traffic across the core network. There is no need for signalling them, so no implementation of LDP or RSVP is provided.
- 4) Internal MPLS tag is used to identify VPN. In VPLS, this tag also identifies the incoming PE, but in SVPLS, as it is based on IP, there is no need for this (source IP is used instead).
- 5) In VPLS, LDP or BGP is used to signal internal MPLS tag (linking MPLS tag with VPN-ID and incoming PE). There are two different working groups, one of them promoting LDP as signalling protocol, and the other promoting BGP. In SVPLS, both mechanisms are avoided and MPLS tag is set to the VPN-ID, so there is no need for signalling it (incoming PE is identified by source IP address). If desired, an upgrade to SVPLS software can be done to include one of these mechanisms.
- 6) In VPLS, a discovery mechanism is suggested although it is not specified. In SVPLS, this mechanism is implemented so new PEs are fast provisioned and VPN information is propagated to all PEs.
- 7) Finally, a keep-alive mechanism is implemented, and so, if one PE fails (or network fails) can be detected and recovered. Moreover, this mechanism is also used to update changed information.

Several improvements can be done to SVPLS in order to continue with this research line. First of all, although it was designed with the idea of working over an IP based core network, a new entity can be added in order to make SVPLS work over a MPLS core. This idea is aligned with the decoupled VPLS initiative introduced in [6] and tested in PREAMBULO (see [2]), where PE functionality is distributed between two entities: C-PE (core side) and U-PE (user side). Moreover, as it is based on IP, security must be improved if core network is not strictly under control. This can be achieved by using IPsec tunnels instead of UDP tunnels although this security can be placed on clients, using end-to-end IPsec.

This service can also be upgraded to support standard VPLS (in order to make it interoperable with commercial solutions) or even support both VPLS standards (BGP signalled and LDP signalled) at the same time (dual PE).

Regarding the performance, it has been shown that with a very basic PC platform, 40Mbps per interface can be

achieved. This can be easily improved with a better hardware, but some effort may also be spent on the development of the software using other capture/generation mechanism than Linux libpcap.

#### REFERENCES

- [1] PREAMBULO: Prototype of high performance multiservice network, based on IPv4/IPv6 over wavelength division multiplexing (TIC2000-0268-P4-C03-01). URL: <http://www.it.uc3m.es/preambulo>
- [2] Valera F., C. García, A. Azcorra, L. Bellido, D. Fernández, J. Berrocal, L.M. Díaz-Vizcaino, J.L. Peña, I. Cabello: Layer 2 VPN experiences over a metro IPoDWDM network. EUNICE 2004, 10th European Summer School and IFIP WG 6.3 Workshop. ISBN 952-15-1187-7 (2004) 38-45
- [3] E-NEXT. Network of Excellence Emerging Network Technologies. <http://www.ist-e-next.net>
- [4] Martini L., E. C. Rosen, G. Heron, N. El-Aawar. Encapsulation Methods for Transport of Ethernet Over MPLS Networks. URL: <http://www.ietf.org/internet-drafts/draft-ietf-pwe3-ethernet-encap-09.txt>
- [5] Kompella K., Y. Rekhter Virtual Private LAN Service. URL: <http://www.ietf.org/internet-drafts/draft-ietf-l2vpn-vpls-bgp-05.txt>
- [6] Lasserre, M., V. Kompella. Virtual Private LAN Service over MPLS. URL: <http://www.ietf.org/internet-drafts/draft-ietf-l2vpn-vpls-ldp-06.txt>