TRANSPORTING ETHERNET SERVICES IN METROPOLITAN AREA NETWORKS (MANS)

Gianmarco Panza' Cecilia Nenov, Glauco Bigini
CEFRIEL Network Systems Unit – Via Fucini 2, 20133 Milan (MI), Italy
Fabio Valente
ALCATEL – Via Trento 30, 20059 Vimercate (MI), Italy

Abstract - In many big cities in the industrialized world, large dedicated data transport networks (the MANs, Metropolitan Area Networks) have been deployed in the last few years to provide cheap broadband Internet access to citizens, as well as broadband interconnections between different branches of a business company. The current MANs are based, in the backbone side, on legacy TDM (Time Division Multiplexing) techniques, like SONET (Synchronous Optical Network) and SDH (Synchronous Digital Hierarchy). The enormous diffusion of the Ethernet technology (wired 802.3 or wireless 802.11) in the LAN environments and the availability of Ethernet interfaces with a very low cost has almost forced telecom operators to consider it as the only possible MAN access technology to sell Internet access services to the large public and business. The MPLS protocol has been chosen by the Metro Ethernet Forum (MEF) [1] to adapt Ethernet traffic to the SONET/SDH networks. This article aims to characterize the behaviour of the different solutions for transporting Ethernet services in the MANs, with or without MPLS support, in terms of efficiency, delay, traffic Engineering capability, fault tolerance, scalability, cost and complexity.

1. THE ETHERNET PROTOCOL

The Ethernet standard (IEEE 802.3) [6] is a physical layer and MAC layer transport protocol designed in the 70's for supporting cheap Local Area Network (LAN) applications. The physical layer defines the type of media and modulation schemes and allows transmission speeds from 10 to 1000 Mbit/s. The MAC layer defines a packet switched transmission mode, based on the so called frames, with a concurrent access to medium and collision control based on the CSMA/CD schema.

The number of concurrent stations and the maximum distance between two stations depends on the transmission speed. For the 10baseT standard (10 Mbit/s full duplex over copper twisted pair), at most 1024 stations are allowed over a single medium with a maximum distance of 4 km between two stations.

The Ethernet frame has a maximum total length of 1526 bytes (octets) including a 22 byte header, a 1500 bytes payload field and a 4 bytes error check code (the FCS). Every station is identified by a 6 bytes MAC address, the MAC layer discards all the frames that are not addressed to the station MAC address.

The Ethernet protocol in its original specification lacks in virtual circuit and in Quality of Service (QoS) support and thus is not natively suited for medium to large scale networks and legacy PSTN services. Subsequent evolutions and enhancements to the base protocols have extended its capabilities including Virtual LAN support (IEEE 802.1Q), fault protection (the Spanning Tree Protocol) and even basic QoS control and MAC layer authentication (IEEE 802.1x).

1.1 The Spanning Tree Protocol (STP)

To avoid the collision effect when many Ethernet station access the same medium, every Ethernet based LAN makes use of Ethernet switches. An Ethernet switch is an active MAC layer device, which has an Ethernet interface for every connected station and automatically builds a MAC forwarding table, to forward to every interface only the frames that are directed to the connected MAC address. With an Ethernet switch the maximum distance between switch and station is only limited by the electric or optical attenuation, since on a station-switch link the medium is not shared by any other station. The Ethernet switches can be connected together to build an all-Ethernet based LAN. To avoid loops in complex Ethernet architectures and to give some kind of fault protection, the Spanning tree protocol has been introduced to transform the physical architecture into logical tree architecture. The STP [10] operates when the network is first set up by flooding the network of control packets to elect the root node of the tree and, starting from it, to disable links that can lead to loops and build a complete tree of the network. A tree reconfiguration message is sent by the root node when a failure is detected by one of the switches and reported to it. The tree configuration can last more than 60 seconds to complete in complex networks and during the reconfiguration all the stations are not allowed to transmit. The more recent evolutions of the STP are the Rapid and Multiple STP (IEEE 802.1w [10] and IEEE 802.1s [10] respectively). The first one is an optimized version of the STP dedicated to switched Ethernet network (without hubs), introducing new type of fault detection and protection to speed up the root election mechanism. The 802.1s allows the simultaneous creation of multiple spanning trees, potentially one for every VLAN, or even backup spanning trees, which can substitute the current spanning tree in case of fault without needing to rebuild a new one.

1.2 The IEEE 802.1q Standard

A lack of the early Ethernet MAC layer was the ability to define virtual LANs. In fact every station connected to an Ethernet network is able to contact every other using the broadcast MAC address. The entire frame directed to the broadcast MAC are delivered to all the connected stations. To be able to split the broadcast domains and to group Ethernet stations, the 802.1Q [10] standard (subsequently included in the 802.1D specification) introduced a small 16 bit tag in the Ethernet header that identifies a group of stations in the Ethernet network. The tag is compiled by the Ethernet switches and

0-7803-8783-X/04/$20.00 © 2004 IEEE

configured by the operator depending on the physical port, the MAC address or even the IP address of the station. Even stations connected to different switched may have the same 802.1Q tag, thus building a Virtual LAN (VLAN). The broadcast frames are delivered only to the stations belonging to the same VLAN of the frame originator, and the stations outside the VLAN are not visible at MAC layer, just like if they were physically separated. The 802.1Q can be used in complex and large networks to identify virtual channels.

2. THE SONET/SDH PROTOCOL

The SONET/SDH protocol has been developed by the ANSI (SONET) [7] and the ITU-T (SDH) [8] for transporting data over high bit rate optical fibre based backbone networks. It supports TDMA synchronous transmission of high speed data flows and offers an advanced fault protection mechanism and a reliable add/drop function, to add and extract low bit rate traffic from a high bit rate aggregate. The SONET and the SDH protocols define a hierarchy of "levels". Each level includes a maximum transmission speed, a frame format and a medium. The levels for SONET are called STS-n if transmitting over copper medium and OC-n if transmitting over optic fibre, the SDH levels are called STM-n for both optical and electric domain, and the SDH STM-1 equals the SONET STS-3 or OC-3. For each level a payload rate, a line rate and an overhead rate is defined: the line rate is the theoretical maximum speed of the line, the payload rate is the effective transmission speed of the data and the overhead rate is the transmission speed of every control and O&M information transfer. SONET/SDH defines a 125 microseconds frame with a matrix internal structure. The frames contain a highly flexible addressing schema that defines a set of identifiers to address many different traffic flows into the frame. This allows multiplexing many different virtual circuits into a single aggregate to allow adding/dropping flows when needed. The circuit identifier length grows when the basic flow is smaller, because a higher aggregation must be done to adapt them to flow into a higher bit rate aggregate. The SONET/SDH protocol is optimized for transporting circuit switched data. A set of adaptation layers are available to transport the most common packet switched traffics like IP/PPP, ATM and MPLS. A protocol has been defined to allow transporting generic packet switched traffic over SONET/SDH, the GFP – Generic Framing Procedure [9], but is currently not widely used due to its complexity.

3. THE MPLS PROTOCOL

The main disadvantage of the IP and Ethernet networks is that to forward packets to the correct destination every traversing node (switch or router) has to check the destination address inside the packet, look up a forwarding table and forward the packet on the correct interface. This mechanism is not optimized in Metropolitan Area Network (MAN) environment, where large virtual LANs are deployed and thus many flows coming from different IP addresses may belong to a single VLAN flow. The MPLS protocol [11] is a Layer 2.5 (between layer 2 and 3) protocol, introduced by IETF to overcome the limitation of IP in routing packets and speed up the forwarding process. MPLS introduces a small header between the L2 and the L3 headers, including a label that identifies the packet flow, instead of the sender station. The MPLS header has been defined for all the mostly used layer 2 protocols, meaning PPP, Ethernet, ATM and Frame Relay. All the MPLS-enabled backbone devices in the network can thus discriminate a flow by only looking into the MPLS header; in addition to this, non-IP networks may take advantage of the MPLS capability of building up virtual channels starting from the IP tables of the router's located at the border. MPLS allows the creation of virtual circuits with a similar flavour of ATM and Frame Relay, but over an IP network, using the IP tables of the backbone for building up the circuits. Like these technologies, MPLS includes traffic engineering capabilities and fault protection techniques. It allows creating virtual circuits through the use of the LSP (Label Switched Path), and allows stacking and chaining labels, to get flow aggregation like in the SONET/SDH networks and allows point to point transmission, as well as, point to multipoint (multicast).

4. TRANSPORTING ETHERNET TRAFFIC IN A MAN ENVIRONMENT – THE EOMPLS PROTOCOL

Starting from the ubiquity of the Ethernet protocol, that is, simply speaking, the only LAN transport protocol in use today (there are still some token ring and ATM based LANs, but have a very small weight in the whole LAN market), an important need in building MAN applications is to have an efficient and reliable transport of the Ethernet frames. What users want today is to connect their PCs Ethernet port or their switches to a "backbone Ethernet plug" and have their frames delivered at destination.

We must note that Ethernet has still some lacks especially regarding traffic engineering and QoS control that still do not allow it to be deployed as a stand alone MAN transport protocol.

Because of the growing importance of the Ethernet networks in enterprise environments, the IETF introduced a mechanism for transporting Layer 2 Ethernet frames into MPLS packets, called Ethernet over MPLS (EoMPLS) [2][3]. The aim is to reduce the overhead of sending out IP traffic over the backbone, and instead send directly the Ethernet frames. This eliminates the need to install a backbone IP gateway for example in the branch offices of an enterprise company and instead use the cheaper Ethernet switches to connect to a backbone access point which is capable of EoMPLS. The EoMPLS solution is based on a work by IETF called the "Martini Draft" from the name of one of its developers and is currently gaining importance in MAN applications.

Basically, when an Ethernet frame arrives to the input interface of the EoMPLS-based gateway, the gateway adds an EoMPLS tunnel label and a VC label (the second one depends on the 802.1Q VLAN tag of the incoming frame) and sends the frame

over the MPLS network. The tunnel label is used to build a tunnel LSP (T-LSP) used to route the traffic from the ingress gateway towards the egress gateway. The VC label is used only inside the ingress and egress gateways to identify the correct VLAN. Many VC labels can be used inside a single T-LSP, depending on the number of VLANs that must be transported.

To foster the use of Ethernet in MAN environment, the Metro Ethernet Forum (an open consortium of Ethernet device manufacturers), adopted the MPLS protocol as a de facto complementary standard to the Ethernet protocol for deploying Ethernet based MANs.

The new Ethernet based MANs must face with the existence of large already deployed SONET/SDH or ATM based MANs, and with the fact that these technologies offer a much higher QoS control, fault protection and error control, and that very unlikely telecom operators will switch from a robust backbone to a nearly experimental all-Ethernet based network. The basic idea is to deploy EoMPLS based MAN access points, that are reached by standard Ethernet equipment installed in the user's home or office, and then transport the Ethernet traffic towards destination using either the robust and reliable SONET/SDH networks or a novel infrastructure based on the last evolutions of Ethernet (e.g. 802.3z and 802.3ae).

5. The Simulation Scenario

The issue of transporting the Ethernet services in MANs concerns the evaluation of important factors, such as the relatively large overhead introduced by the encapsulation of different transport protocols and the end-to-end delay which is influenced by this overhead. In order to quantitatively characterize the different transport solutions in terms of overhead and end-to-end delay we made a complete set of simulation using the OPNET network simulation tool. We considered three different situations: a native Ethernet MAN and the deployment of EoMPLS over Ethernet or SONET/SDH. We identified a MAN scenario with 3 intermediate switching points, using the three different transport solutions, plain Ethernet, EoMPLS/Ethernet, and EoMPLS/SDH.

The audio source model is a classic ON-OFF model that is valid for every source coder/decoder since it models the human voice instead of codec behaviour. The model characterizes the voice in terms of two variables, Kon and Koff, which express the activity and inactivity periods of the source. We used two codecs to (de) compress data coming from the source: a G.729 using 20ms packetization, which means 2 G.729 frames per packet, which coincide with a 20 bytes payload and 40 bytes header (IP+UDP+RTP), and a G.711 using the same packetization time that leads to a 160 bytes packet.

The video source model uses a real H.263 video trace read from a file, with 256 kbit/s coding. The maximum frame rate is 25 fps with inter-frame delay of 40 ms. We did choose two different packetizations, one at 576 bytes and the other at 1400 bytes payload per packet. A complete frame is packetized and sent when completely available in the sender buffer. The fixed overhead introduced by IP/UDP/RTP, as in the case of the audio source is 40 bytes per packet.

Scenario	Source	Payload
A1 Audio	G.711	20 Byte
A2 Audio	G.721	160 Byte
V1 Video	H.263	576 Byte
V2 Video	H.263	1400 Byte
AV	Any	Any

Table 1 - Simulation scenarios

We ran 5 different simulations, including the various sources explained before. The 5 scenarios are listed in the table above, and show the different ingress traffic types. In the AV scenario all the audio and video traffics are sent together inside the network. Every network node is traversed by a background traffic built of video and audio packets.

6. The Overhead Problem

The three transport solutions have different overhead given by the packet and frame encapsulation.

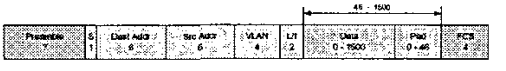

Figure 1 - The Ethernet frame format

The overhead introduced by Ethernet is 30 bytes per frame (including the 4 bytes VLAN tag), with variable payload length. Using the Plain Ethernet transport solution the complete overhead consists of the fixed 30 bytes overhead plus the time between a just sent frame and the following that can be taken into account as another 12 bytes, that leads to a 42 bytes complete overhead.

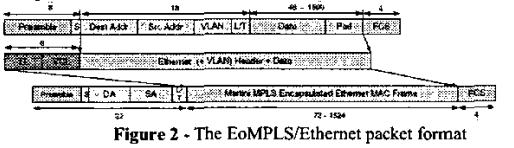

Figure 2 - The EoMPLS/Ethernet packet format

In the second solution (EoMPLS/Ethernet) we must consider that when encapsulating Ethernet frames in MPLS packets, we should not count the preamble and the FCS, that are included in the transport network frame. This thus leads to an overhead of 18 bytes for the inner Ethernet frame with VLAN tag, an 8 bytes EoMPLS header and a final complete 26 bytes Ethernet header (without VLAN tag). The complete header is 64 bytes per IP packet sent.

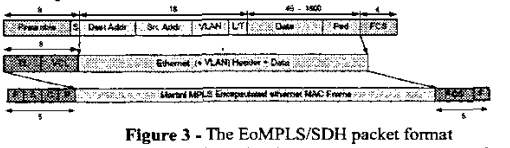

Figure 3 - The EoMPLS/SDH packet format

In the third case, keeping the inner Ethernet L2 header and the MPLS header, we have to add the PPP header (in this case EoMPLS is adapted to SDH using the PPP protocol

encapsulation), which is 10 bytes and the SDH header that has not fixed length, but is approximately 1/27 of the transported payload length. The total overhead in this case varies from a minimum of approximately 40 bytes to a maximum of 93 bytes, calculated as 18+8+1/27*(header + the minimum payload of 46 bytes and the maximum 1500 bytes).

In the charts below we can see the percentage of overhead compared with the Ethernet frame and with the Payload data (the IP packet) with different packet sizes. As we can see, the EoMPLS/SDH solution has significantly smaller overhead than the EoMPLS/Ethernet solution when dealing with smaller packets, similar to the Plain Ethernet solution. The overhead difference shrinks when the packet size grows, due to the percentage-based overhead in SDH, getting down to almost zero with 1000 bytes packets.

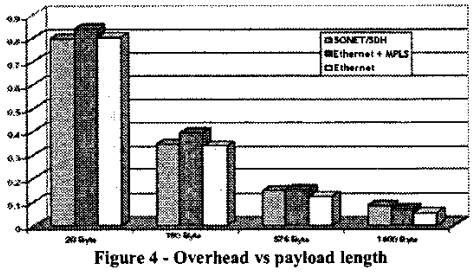

Figure 4 - Overhead vs payload length

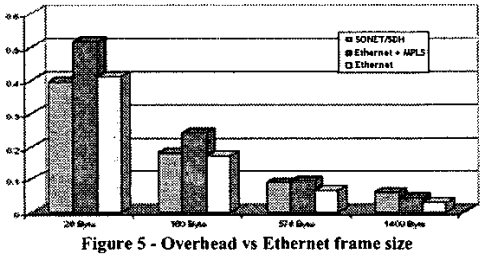

Figure 5 - Overhead vs Ethernet frame size

7. THE END-TO-END DELAY PROBLEM

To have a quantitative comparison between the different transport solutions in terms of end-to-end delays, we conducted various simulations.

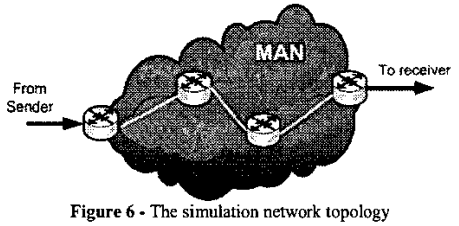

Figure 6 - The simulation network topology

The network topology we used is very simple, but still typical for many Metropolitan Area Networks, with 2 ingress nodes and two transit nodes.

The end to end delay has been characterized using two components: the processing delay in every node and the transmission delay. The processing delay has been derived from the values specified in the data sheets of many products from different vendors and has been assigned as in the following table

	20 Byte	160 Byte	576 Byte	1400 Byte
Router	7 µs	9 µs	14 µs	24 µs
L2 Switch	8 µs	11 µs	17 µs	19 µs

Table 2 - Switching delays

The transmission delay depends on the link capacity and on the packet length and ahs been modelled in the simulations.

In the first graph we see the results of the A2 Scenario (see before), with 160 bytes audio packets. We can note that in this case the best performance is obtained using the EoMPLS/Ethernet, shortly followed by the SDH case. The delay is a little higher in the all-Ethernet case. This is due to the higher processing delay introduced by the MAC address switching, that is longer than the MPLS label swapping.

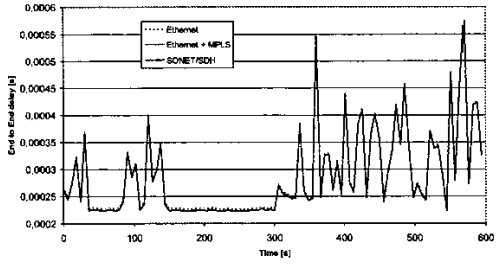

Figure 7 - End to end delay A2 scenario

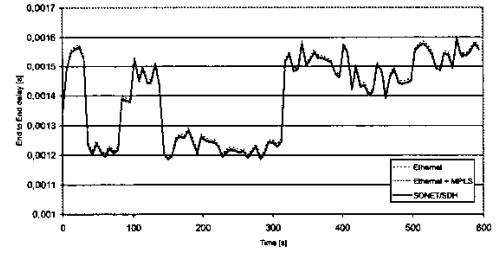

Figure 8 - End to and delay V2 scenario

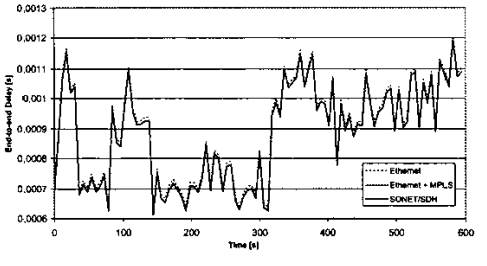

Figure 9 - End to end delay AV scenario

The same happens in the V2 scenario with 1400 bytes video packet and in the AV scenario with all traffic types together. The results show that the MPLS introduction, in spite of the higher overhead, implies lower end to end delay in networks with several traversed nodes, because the smaller processing latency of the MPLS fixed-length Label overcompensate the higher transmission delay of bigger packets. Another result is the small advantage of the EoMPLS/Ethernet compared with the EoMPLS/SDH solution because of the smaller overhead of the first one in case of large packets (V2 case).

The following table shows the complete results of this simulation, with the average end to end delays in the different cases. The results are obtained using two types of interfering traffic, one with 160 byte audio packet plus 572 byte video traffic and the other with the same audio and 1400 byte video.

Scenario	160B+576B			160B+1400B		
	Average (μs)	Δ(*)	Δ(°)	Average	Δ(*)	Δ(°)
All-Ethernet						
A1	135	±8.69	±13.6	170	±15	±23.5
A2	280	±0.02	±0.03	297	±11.5	±18.1
V1	950	±0.02	±0.03	966	±0.02	±0.04
V2	1404	±0.02	±0.03	1419	±2.64	±4.15
EoMPLS/Ethernet						
A1	138	±10.9	±17.1	162	±15.2	±23.8
A2	275	±8.91	±14	292	±11.6	±18.3
V1	942	±0.02	±0.03	958	±0.02	±0.03
V2	1392	±0.02	±0.04	1407	±0.02	±0.04
EoMPLS/SDH						
A1	154	±10.9	±17.1	171	±15.2	±23.8
A2	278	±8.91	±14	294	±11.6	±18.3
V1	944	±0.02	±0.03	960	±0.02	±0.03
V2	1396	±0.02	±0.04	1411	±0.02	±0.04

Table 3 - End to end delay simulation results

The EoMPLS/Ethernet and EoMPLS/SDH solution are substantially equivalent, with a small margin in favour of the former. The all Ethernet shows worse results due to the longer frame switching time. The results let us conclude that the introduction of MPLS over an Ethernet based MAN allows reaching almost the same performance as in the case of SDH, in terms of end-to-end delay.

8. THE TRAFFIC ENGINEERING PROBLEM

One of the main concerns about building an all-Ethernet MAN is related to the poor performance of the QoS control mechanisms provided by this protocol. Ethernet relies in fact only on the STP to build a network tree and to distribute traffic. Traffic engineering capabilities of MPLS allow to dynamically distributing traffic on the network links to optimize resources utilization and indirectly to reduce delay and losses. The problem is that the STP calculates a single delivery tree with consequent limited load balancing. A tree recalculation is issued only in case of failures.

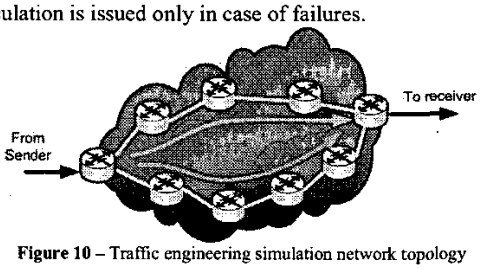

Figure 10 – Traffic engineering simulation network topology

To understand this advantage, consider the graph below, obtained from a simulation on the simple scenario, just taken as a meaningful example, shown in Figure 10. We simulated an all-Ethernet and an MPLS network with background link load of 50% (links are all 10 Mbit/s) and sent into the network two flows of 2 Mbit/s video traffic. Using MPLS the path cost could be calculated as a function of the link loads and the packet forwarding is decided basing on this cost. When sending the flows, in the Ethernet case, the already built tree has selected path A because of the smaller number of transit nodes, while MPLS dynamically can take into account of the network load and select alternative paths.

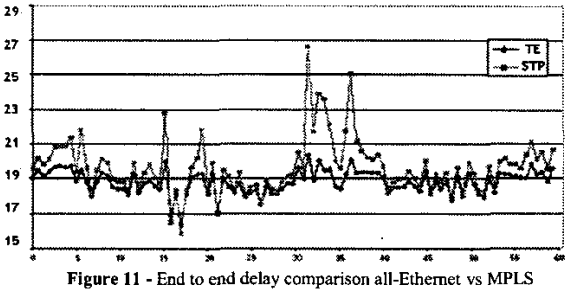

Figure 11 - End to end delay comparison all-Ethernet vs MPLS

As we can see the introduction of MPLS greatly reduces the end-to-end delay variation (jitter), giving a better control on the QoS than the simple Ethernet protocol, even in a very simple network topology like this.

For what concerns the SONET/SDH based solution, native TE functionality of such technology can be also exploited, mainly in a static manner due to configuration issues. However, to achieve a greater flexibility, MPLS capabilities have to be employed.

9. THE FAULT PROTECTION PROBLEM

Ethernet protocol has no other way to protect the network from faults than using the Spanning Tree Protocol, which takes too long to rebuild a whole delivery tree when a fault is reported from a switch. Using the Rapid STP (802.1w) reduces the tree building time from tens of seconds or more to 1 or 2 seconds. The 802.1s is not a speed improvement, but only a flexibility enhancement, to support the VLAN management, thus recovering an Ethernet network from a fault would require, in

the best case, 1-2 seconds for Rapid STP to rebuild the network tree.

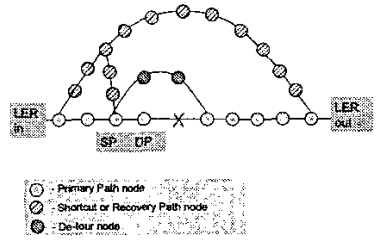

Figure 12 - Application of MPLS fast reroute with either recovery path and shortcuts, or de-tour path (LER - Label Edge Router, SP - Switching Point, DP - Detection Point; l =1, N=3)

The introduction of the MPLS protocol between layer 2 and 3 eliminates the need to use any fault protection mechanism provided by the lower layer. MPLS has its own protection schemas that have been developed with the specific aim to reach the same fault protection level of SONET/SDH, that is, within 60 milliseconds, where 10 ms are for the fault detection and 50 ms are for switching to a backup circuit. This objective is driven by the fact that the SDH networks have been developed to support telephone network traffic, the most demanding in terms of reliability (today's telephone network have reliability levels of more then 99.99999%).

While the switching time on the backup circuit is determined by the MPLS mechanisms, the fault detection time is still given by the underlying layer 1 and 2 technologies. There are in fact 2 types of faults: link fault and node fault (the entire node is not responding). The detection times are different in the two cases. As previously said, in a SDH network the link fault detection time is about 10 ms, while in an Ethernet network is about 16 ms (time frequency of dedicated signal for the purpose). The node failure detection time is instead similar in the two cases and can reach various tens of milliseconds.

After the fault has been detected, an alternative path must be taken to bypass the fault. This is done by MPLS with various techniques. The most widely accepted is called Fast Reroute [5]. In fast reroute, the node that detects the fault simply resends the traffic back to the preceding node without sending any signalling; the preceding node forwards the reflected traffic back in the path and so on, until it reaches a switching node which has an alternative path configured. When this node detects the returned traffic, it stops sending traffic on the primary path (the working path) and reroutes it on the recovery path. To optimize the switching time, a complete recovery path is configured a priori and all the designated switching nodes of the concerned primary path have shortcuts to it. If the recovery path and the shortcuts are not configured, the issued switching node will set them run-time using standard MPLS signalling. Another possibility is to consider a de-tour path for each failure along the working path that simply goes around the fault.

Assuming that the fast reroute mechanism is used, we can model the total MPLS fault recovery time as follows: Trec = Td+Tr, where Td is the fault detection time and Tr the recovery time, defined as the interval between the detection of the fault and the switching to either the recovery path or the de-tour path. Td is 16 ms as stated before in the case of Ethernet and 10 ms in SDH.

It can be shown [5] that the recovery time Tr is 3LTcross+(2N-1)Tp. Where, L is the number of nodes from the fault detection point to the switching point and N is the total number of nodes constituting the shortcut or the de-tour path. In the second term of the equation Tp is the processing time needed to process (and transmit) the MPLS signalling, and is about 10ms; this term is simply about 1 ms if the shortcut or the de-tour path has already been configured. Tcross is the time needed to cross a node. It can be shown that due to buffering delays the time to switch from the working to the recovery path after the shortcut or the de-tour path establishment is 3LTcross; instead, if no packet reordering is provided, the nodes do not buffer traffic and hence such a time is LTcross. Tcross can be assumed 10 ms when the reflected traffic is treated with high priority (the common case. In a well configured MPLS network, N and l are no more than 3. In the former mentioned case of fast-reroute, the recovery path and the shortcuts are typically already configured, thus to the first term of the Tr expression, about just 1 ms of further delay should be added.

A total count of the recovery time, in a worst case, leads us to Trec=16+9*10+5*10=166 ms. A typical case, where the detour path or the recovery path together with the related shortcuts have been already configured, we have Trec equals to 10-20 ms plus 90 or 30 ms (at worst, with l=3) if the detection and recovery points do not coincide, respectively with or without node buffering.

This analysis shows that, if the network is well configured, the recovery mechanisms developed for MPLS networks reach the target of SDH-like performance. Thus an EoMPLS/Ethernet solution can be seen as nearly equivalent to an all-SDH solution in terms of fault protection and reliability. Finally, for an EoMPLS/SDH solution the results are extremely similar.

10. THE SCALABILITY PROBLEM

Scalability is a matter of size of the network. SDH is a protocol that has been designed for large scale optical networks, but is not well suited due to its cost to small networks, where Ethernet reigns without opponents. On the other side, the Ethernet protocol is not sufficiently scalable for being used in large networks such as MANs.

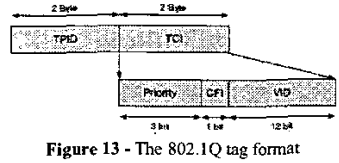

Figure 13 - The 802.1Q tag format

Starting from the virtual LANs, Ethernet supports this type of services through the 802.1Q standard, but since the 802.1Q tag is only 4 bytes long, with only 12 bits used for actual VLAN identification (the VID field in Figure 13), there is an upper limit of 4096 different virtual LANs for every MAN. Another scalability problem introduced by Ethernet is the relatively slow spanning tree protocol which introduces a high computational overhead in the network nodes, and the fact that currently available Ethernet switches have not been designed to work in a complex and large scale environment, with a great number of MAC addresses to handle. The introduction of MPLS overcomes these limitations. At first an MPLS label is 20 bits long, thus targeting 1 million virtual paths, much more than the 802.1Q standard. Furthermore, MPLS allows other scalability features such as label stacking and increases the switching speed due to the small size of the labels. Metro Ethernet Forum suggests the introduction of an additional tag to improve scalability in all-Ethernet networks. This solution (Ethernet Multiplexing Function) partially solves the scalability problem, but introduces interoperability issues and an increased complexity of the Ethernet hardware to be deployed.

11. The Cost and Complexity

The SDH devices are far more complex than the Ethernet switches, still being more stable and reliable. The same situation applies to cost considerations, where SDH devices greatly overtake Ethernet hardware. Even the protocol stacks in using the EoMPLS/SDH solution are more complex because of the greater number of adaptation layer needed to transport the Ethernet traffic over the SDH infrastructure.

There is even another issue when SDH is used in MAN environments, where a large number of small virtual LANs are deployed, which also deals with scalability concerns: the fragmentation of bandwidth. This problem, which has partially been solved in the next generation SDH networks, happens because SDH has not been designed for supporting fine grain traffic engineering, and wastes bandwidth when too many low bandwidth channels are allocated inside a large capable backbone. The solution that considers Ethernet as the transport infrastructure is of course simpler and cheaper, especially from the customer equipment side.

Therefore, the upper adaptation layer of EoMPLS and MPLS of course, with a small cost overhead greatly reduces complexity while increasing scalability.

12. Conclusions

This paper mainly analyzes the performance of different solutions for transporting Ethernet services in MANs based on EoMPLS either over SONET/SDH or over Ethernet, quantitatively characterizing the overhead and encapsulation/decapsulation delays, as well as QoS capabilities in general, and to identify the weaknesses of this technology. The EoMPLS over SONET/SDH or over Ethernet has also been compared with a full-Ethernet MAN solution, which considers Ethernet as the only transport protocol inside the MAN. Moreover, a comparison has been made between the different fault protection mechanisms that can be exploited in the above analyzed MAN technologies.

What we have obtained as a result is that the full Ethernet solution is obviously not well suited for deployment in a large MAN, since it doesn't offer in any way traffic engineering capabilities, sufficient QoS control mechanisms, scalability and flexibility in assigning and maintaining VPN channel that are mandatory in such environments. The EoMPLS either over Ethernet or over SONET/SDH offers much better results. In particular, the first approach allows to dramatically reduce the cost and complexity of the backbone equipment (IP/MPLS routers with standard Ethernet interfaces) and of O&M, still making use of the advantages of MPLS for QoS functionalities and VPN support, while the last one is much more expensive, but shows typically a lower overhead (the SONET/SDH frame header is smaller compared to the Ethernet header for nearly all packet sizes, up to about 1400 bytes), a greater fault tolerance and reliability and has the advantage of reusing the already existing SONET/SDH infrastructure (natively deployed to transport PSTN telephone traffic).

The large diffusion of cheap broadband MAN access is the starting point for the successful deployment of value added services. The SONET/SDH and the Ethernet are mature technologies: a good candidate for a contact point between these two different worlds needed to build robust, cheap and reliable Metropolitan Area Network is the MPLS protocol. We may conclude, on the basis of the results obtained in this study, that the EoMPLS over Ethernet is the best solution for new MAN networks; while the EoMPLS over SONET/SDH can actually be considered as the only option for legacy reasons and leveraging the investments, where a SDH MAN infrastructure already exists.

13. References

[1] **Metro Ethernet Forum web page** http://www.metroethernetforum.org/

[2] **Luca Martini et al** - *Transport of Layer 2 Frames over MPLS* – **IETF Internet Draft**

[3] **Luca Martini et al** - *Encapsulation Methods for Transport of Ethernet Frames over IP/MPLS Networks* – **IETF Internet Draft**

[4] *Framework for Multi-Protocol Label Switching (MPLS)-based Recovery* – **IETF RFC 3469**

[5] **L. Hundessa, J. Domingo-Pascual** - *Reliable and fast rerouting mechanism for a protected label switched path* – **Proceedings IEEE Globecom 2002**

[6] **IEEE** - IEEE 802.3 CSMA/CD (ETHERNET) - http://www.ieee802.org/3/

[7] **ANSI** – *T1.105: Sonet* – www.ansi.org

[8] **ITU-T** – *G.707, G.781, G.782, G.783, G.803: SDH* – www.itu.int

[9] **ITU-T** - *IP over SONET/SDH ITU-T G.7041 Generic Framing Procedure* – www.itu.int, **June 2002**

[10] **IEEE** – *IEEE 802.1series page* - **http://www.ieee802.org/1/**

[11] **IETF** – *MPLS Multiprotocol Label Switching* - **http://www.ietf.org/html.charters/mpls-charter.html**