

A New Scheme for Interconnecting LANs with Label Switching Bridges

Thierry K. Feuzeu, Bernard Cousin

▶ **To cite this version:**

Thierry K. Feuzeu, Bernard Cousin. A New Scheme for Interconnecting LANs with Label Switching Bridges. 30th Annual IEEE Conference on Local Computer Networks (LCN)., IEEE, Nov 2005, Sydney, Australia., France. pp.303-310, 10.1109/LCN.2005.12 . inria-00084502

HAL Id: inria-00084502
https://inria.hal.science/inria-00084502

Submitted on 7 Jul 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Scheme for Interconnecting LANs with Label Switching Bridges

Thierry K. Feuzeu and Bernard Cousin
IRISA/INRIA
Université de Rennes I – Campus de Beaulieu
35042 Rennes Cedex, France
{tfeuzeuk,bcousin}@irisa.fr

Abstract

Ethernet, which has traditionally been the dominant technology in Local Area Networks, is now facing new challenges due to the fact that networks have scaled and today's applications require more bandwidth and increased robustness against failures. The limitations of Ethernet are mainly imposed by the Spanning Tree Protocol*, which is necessary to prevent loops and provide redundancy in a switched network. The successful deployment of Ethernet over MPLS in the metro area have proved that Ethernet LANs can be further enriched with the services enabled by the label switching technique. In this article, we propose a new LAN interconnection technique based on label switching, which enhances network services, simplifies switch functionalities, while remaining flexible enough to be easily deployed in most LAN environments.*

1. Introduction

Ethernet is the most popular LAN technology in use today due to its high performance, its ease of use and maintenance. Enriched with VLAN capabilities, Ethernet switching offers a way to further enhance network services, by easing adds, moves, changes, and increasing performances. An Ethernet switch interconnects multiple LAN segments, and confine traffic only to those segments the destinations are connected to. The switched LAN distance coverage is then extended, since each segment has available to it the maximum distance allowed by the technology. The *Spanning Tree Protocol, STP* ([11]), is used in a switched LAN to prevent loops in the interconnection, and provide alternative paths if a link or switch fails.

As a result of its successful evolutions, Ethernet has now entered various other areas, such as metro networks. Also, LANs have scaled and today's applications need more bandwidth and reliability. It is then a strong requirement that LAN technologies must offer greater extensibility, performance and robustness against failures. The STP, which is used to prevent loops and provide redundancy, imposes several limitations to switched LANs. First, it provides a convergence time of 30 to 50 seconds, which is not acceptable for today's applications. Even with improvements like the *Rapid STP* and *Multiple STP*, the convergence time can grow up to a few seconds. Then, because it blocks links to prevent loops, it can lead to inefficient use of network resources. Although this is not a problem in most LAN where the bandwidth is low cost and over provisioned, blocking a WAN or MAN link may have greater impact on the overall network performance.

In order to overcome Ethernet switching limitations, the idea of combining Ethernet with label switching technique has been investigated, first with ATM LAN Emulation ([12]), and more recently with the introduction of Ethernet over MPLS in metro networks ([1, 2]). However, since ATM and MPLS are primarily designed for carrier-grade networks, they do not fit well with the requirements for low cost and simplicity of LANs.

In this paper, we introduce *Laidanet*, an Ethernet LAN interconnection technique based on label switching. The goal of *Laidanet* is to provide Ethernet LANs with the services enabled by the label switching technique, while keeping the switch functionalities as simple as possible, and without changing end stations operation.

The rest of this paper proceeds as follows. Section 2 presents an overview of Ethernet switching and the motivations for our work, section 3 presents *Laidanet* operation and some applications, section 4 discusses its interests, its challenges, and comments experimentation results, and then comes the conclusion.

2. Transparent Bridging

2.1. Principles of Operation

An Ethernet switch interconnects many LAN segments, and confines traffic on the segments where it is destined, rather than wasting bandwidth on other segments.

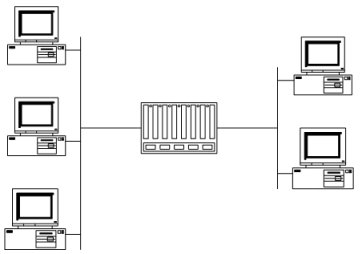


Figure 1: A Switched Ethernet

2.1.1. Frame Forwarding When a frame is received on any port, the switch looks up the destination address in its address table, and determines the port to which that address maps. The frame is discarded if the port on which the frame is received is the same port where the target destination resides. When a switch performs the lookup in the address table, it can happen that the destination address is not in the table. The switch then forwards the frame onto all ports, except the one on which it arrives. Thus, flooding will permit the communication with destinations unknown to the switch, in the event that the destination is present on some switch port other than the port of arrival.

2.1.2. Address Table Generation The switch address table can be built automatically by considering the source address in received frame. The switch performs a table lookup for an entry corresponding to the source address. If an entry is not found, the switch creates a new entry for this newly-learned address, with the port mapping indicating the port on which the frames arrives. If an entry is already in the table, the port mapping is updated. This allows the switch to properly map stations that have moved from one LAN to another.

A switch ages out entries of its address table when the related stations have not been heard for some period of time. When it performs table lookup for the source address, it also flags the entry as being still active. On a regular basis, the switch removes the entries that have not been flagged for some period of time. The aging process helps restricting the entries in the address table to only those stations that are known to be currently active.

2.1.3. The *Spanning Tree* Protocol (STP) The *Spanning Tree* Protocol was defined as the standard loop resolution protocol. It calculates, configures and maintains a tree topology in the switched LAN. A switch is elected as the root of the tree, and for each and every link in the LAN, a single switch is responsible for forwarding traffic from the root to that link. This is the *designated switch*. If the designated switch has two ports on the same link, only one port is elected as the *designated port* for this link. The *Spanning Tree* is completely defined by the set of designated switches and designated ports. All other ports are blocked, and do not forward frames. On a regular basis, switches exchange *BPDU* with their neighbors, so they can recalculate the *spanning tree* if the tolopogy changes.

2.1.4. VLAN A switched LAN defines as a single broadcast domain: broadcasted frames are forwarded to all stations, wasting bandwidth on the whole network. A VLAN ([4, 7, 8]) is a group of end-stations, perhaps on multiple physical LAN segments, that are not constrained by their physical location and can communicate as if they were on a common LAN. VLAN allow a network administrator to logically segment a LAN into different broadcast domains. Broadcasted frames are forwarded only to a those stations which are member of the same VLAN. There are several ways in which VLAN membership can be defined: port grouping, MAC-layer grouping, and network-layer grouping.

VLAN were developed as an alternative solution to using routers to contain broadcast traffic. In addition, VLAN can ease the establishment of virtual workgroups, and automatically track membership if a station moves, or if a workgroup changes. The advantages of VLANs over switched LANs are performance, security, simplified administration and reduced costs.

2.2. Motivations of our Work

2.2.1. Preserve Switched LAN Concepts The original rationale for the development of LAN switches is LAN extension, both in terms of distance and number of stations. A switched LAN offers the possibility configurations that are not available to the shared LAN user. Novel concepts introduced by switching are separate access domains, extended distance limitations, increased aggregate capacity and data rate flexibility.

In a shared Ethernet LAN, the CSMA/CD MAC algorithm ([5]) is used to arbitrate the use of the shared channel, also defined as an access domain or a collision domain. Stations in the same access domain can experience access contention, with the resulting collision

and backoff. In a switched LAN, each switch port terminates the access domain associated with that port. There can not be collisions between stations connected to different ports.

Switches allow to extend the distance coverage of a LAN, since each switch port has available to it the maximum distance provided by the technology.

A switched LAN provides greater data carrying capacity than a shared LAN. In a shared LAN, the LAN capacity is shared among all of the attached devices. Since a switch provides dedicated capacity on each port, the total LAN capacity increases with the number of switch ports.

All devices connected to a given shared LAN must operate at the same data rate, while independant LANs can operate at different data rates. This allow flexibility in deploying stations at different data rates; high performance is provided where needed without burdening all users with the cost associated with a faster LAN.

2.2.2. Overcome Switched LAN Limitations A robust, reliable network needs to transfer traffic efficiently, provide redundancy and recover quickly from faults. In a switched Ethernet network, the STP offers redundant connections and eliminates the danger of data traffic loops. The original STP typically recovers a link failure within 50 seconds. While such an outage was acceptable when the protocol was designed, today's applications (voice and video, for example) require much faster network convergence.

Ethernet switches are also used to interconnect geographically-separated locations using WAN links. While the STP will properly resolve loops in such configurations, some important issues arise from the difference between LAN and WAN technologies. WAN links offer significantly lower data rates, higher error rates and higher costs than LAN links. Consequently, the overhead imposed by the STP constitutes a greater percentage of the available capacity, and blocking a WAN port will have greater impact on throughput and delay.

2.2.3. Take Advantage of Label Switching in Ethernet LANs The idea of combining label switching technique with Ethernet have been investigated with ATM LAN Emulation, and more recently with the introduction of Ethernet over MPLS in metro networks. The goal is to provide end users with the high bandwidth of Ethernet, and further enhance services with the traffic engineering capabilities offered by ATM or MPLS. Ethernet over MPLS is now a strong candidate for deploying metro networks services.

However, ATM and MPLS are primarily designed for carrier-grade networks. They offer much more functionalities than needed in LAN environment, consequently they are too complex and too expensive to be effectively deployed in most LANs.

2.2.4. Separate Bridging from Bridges An original solution was proposed in [3] to overcome the increasing complexity of the Internet routing architecture: clearly separate the routing architecture from the forwarding architecture. The routers simply forward packets (for the most part), and the routing functionality is moved to a *Routing Control Platform* where its complexity can be better managed. Such an architecture simplifies network configuration and preserves backward compatibility. Routes are selected based on high level goals rather than manipulation of complex routing protocol attributes.

The *Laidanet*'s design takes advantage of this paradigm. The network configuration is moved to a control entity, and the switches simply forward frames.

3. Switching in *Laidanet* Networks

Laidanet stands for *Label In Destination Address Network*. It results from an attempt to combine label switching with Ethernet in order to enhance LAN performance and services, while keeping bridge functionalities as simple as possible. A *Laidanet* network is composed by a set of Ethernet hosts, interconnected by a set of label switching bridges, all managed by a *Controller* (figure 2).

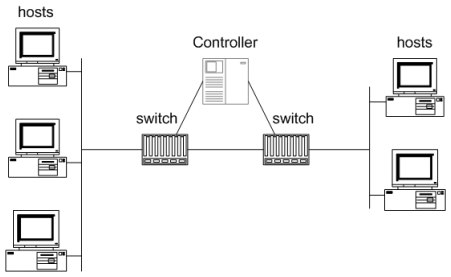


Figure 2: A *Laidanet* Network

The hosts connected to a *Laidanet* network are standard Ethernet end stations, whose behaviour remains unchanged. They run an address resolution protocol to query their correspondant MAC address, and they send the frames in destination of the resulting address.

The switches form the *Laidanet* core. Their forwarding decision is based on the label carried in the frames.

The *Controller* is a logical entity rather than a physical node. It is the central element of a *Laidanet* network: it configures switched paths in the network,

based on the topology and the communication needs of end-stations and switches.

3.1. *Laidanet* Configuration

Because the behaviour of hosts connected to a *Laidanet* network must be kept unchanged, it is a strong requirement that *Laidanet* networks are auto-configurable, and operate transparently.

The figure 3 illustrates how *Laidanet* is configured. The main steps are topology discovery (TD), *Controller* configuration (CC) and switch configuration (SC).

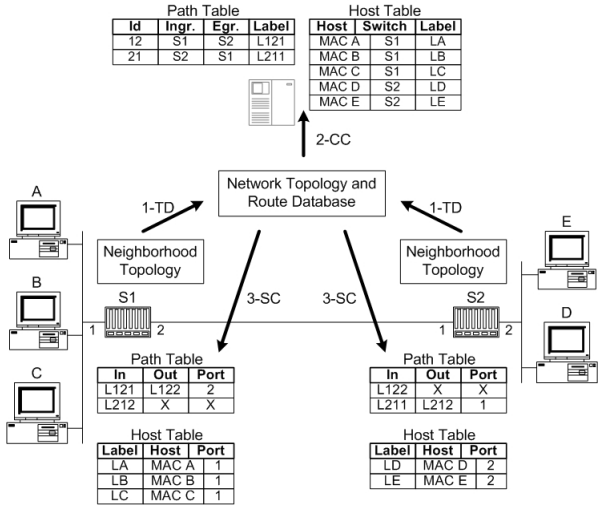


Figure 3: *Laidanet* Configuration

3.1.1. Topology Discovery To compute and configure paths in the network, the *Controller* must maintain an up-to-date representation of the entire network topology. To discover their neighborhood, the *Laidanet* switches exchange topology discovery messages among them, and analyse the frames end-stations send on the network. Each switch reports its local topology to the *Controller*, which is then able to derive the topology of the entire network.

3.1.2. *Controller* Configuration The *Controller* computes and configures a set of paths, based on the network topology, and the communication needs of switches and end-stations. Then, it summarizes this information in two tables: the *Host Table* and the *Path Table* (figure 3).

There is an entry in the *Host Table* for each and every station connected to the network, which indicates the MAC address, the switch the station is connected to, and the label (so called *host label*) identifying the station within this switch. There is an entry in the *Path Table* for each and every unidirectional path configured in the network, which indicates the *ingress* and *egress* switches, and the label (so called *path label*) which causes a frame entering the *ingress* to follow this path.

3.1.3. Switch Configuration The *Controller* also configures a *Host Table* and a *Path Table* in every switch, whose content differs from its one (figure 3).

There is an entry in the *Host Table* for each and every station directly connected to a local port, which indicates the MAC address, the associated *host label*, and the port to which the station is connected. There is an entry in the *Path Table* for each and every path crossing the switch, which indicates the corresponding input label, output label and port. In the *egress* switch of a given path, the output label and port have *null* values, indicating the end of the path.

3.2. *Laidanet* Operation

3.2.1. Address Resolution The figure 4 illustrates the *Laidanet* address resolution mechanism, with focus on the target MAC address field of the request.

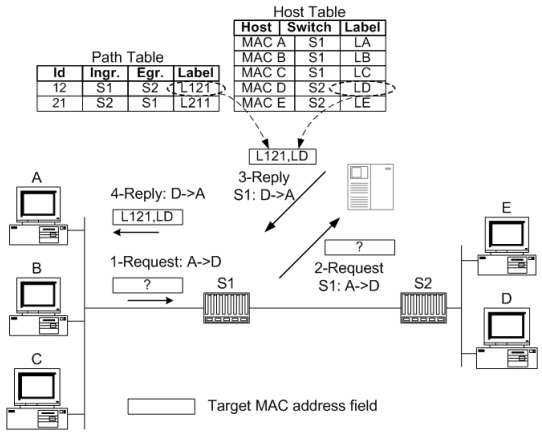


Figure 4: Address Resolution in *Laidanet*

The *Laidanet* switches receive and analyse all address resolution requests. When the target station is not connected to a local port, they forward the request to the *Controller*.

On receiving such a request, the *Controller* first determines the target MAC address, and then looks up its *Host Table* and *Path Table* respectively for the corresponding *host label* and *path label*. In figure 4, the *path label* (*L12*) is the one associated to the path between the *S1* and *S2*, while the *host label* (*LD*) is the

one associated to the station *D* in the switch *S2*. The *Controller* finally returns a special MAC address (so called a *labelled MAC address*) in which the *path label* and *host label* are embedded, rather than the real MAC address of the target host.

3.2.2. Frame Forwarding On receiving a *labelled frame*, a *Laidanet* switch (*S1* in figure 5) looks up its *Path Table* for the entry indexed by the incoming *path label*, writes the indicated output label in the *path label* field, and forwards the frame to the indicated port.

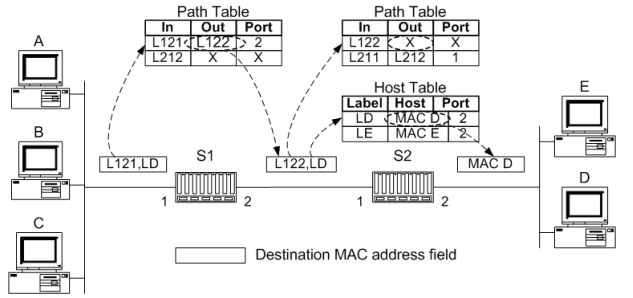


Figure 5: Frame Forwarding in *Laidanet*

In the *egress* switch (*S2* in figure 5), the output label and port have *null* values. The switch then looks up its *Host Table* for the entry indexed by the incoming *host label*, writes the indicated MAC address in the *Destination Address* field, and forwards the frame to the indicated port. The target host then receives a normal Ethernet frame.

3.3. More About Labels in *Laidanet*

3.3.1. Relationship Between Labels in *Controller* and Labels in Switches In *Laidanet*, the *Controller* returns two labels (one from its *Path Table* and one from its *Host Table*) to hosts in address resolution replies. The switches use these labels to forward frame. For *Laidanet* to operate properly, the same two labels must be found as input label respectively in the ingress switch *Path Table* and the egress switch *Host Table*.

3.3.2. MAC address Format In *Laidanet*, the labels are carried in the *Destination Address* field of the standard Ethernet frame. Thus, the *Laidanet* frame format is the same as the Ethernet frame format, while the MAC address format is different.

The 24 bits length *Organizationally Unique Identifier (OUI)* has a constant value which distinguishes *labelled frames* from other Ethernet frames. The labels are carried in the remaining 24 bits.

3.3.3. Label Space and Forwarding Table Size Given p the length of the *path label*, h the length of the *host label*, and the hypothesis that each switch has two labels allocated for each pair of interconnected LAN, a *Laidanet* can connect $2^{\frac{p}{2}}$ LANs with 2^h hosts each. The maximum table size in a switch is 2^p+2^h. Minimising the table size leads to $p = h$, hence both *path label* and *host label* are 12 bits length, for a maximum table size of 8192, which allow up to 262144 hosts to be connected to a *Laidanet*.

3.4. Additional Services

Thank to the use of label switching, explicit routing can be easily implemented in *Laidanet*, as well as more efficient fault protection techniques, with very few (and even no) extra processing or traffic in switches.

3.4.1. VLAN Implementing VLAN in a *Laidanet* network only requires modifications to the *Controller*. A VLAN column is added to the *Host Table* and *Path Table*, and filled according to the VLAN membership rules defined by the network administrator. The *Controller* will send a reply to an address resolution request only if the source and target hosts are members of a same VLAN.

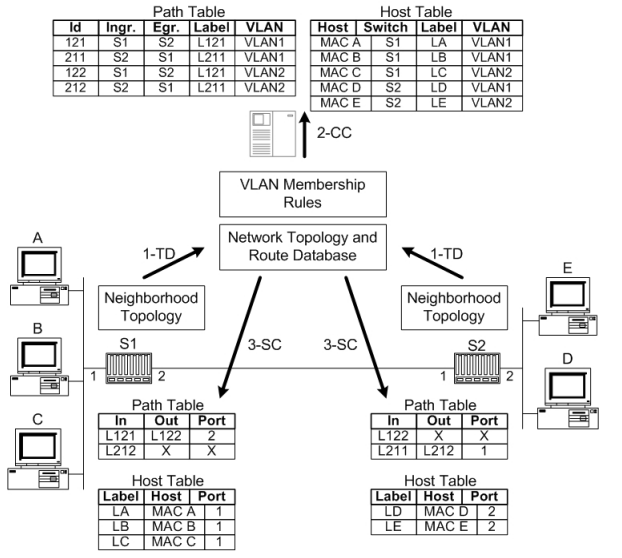


Figure 6: VLAN Implementation in *Laidanet*

VLAN implementation in *Laidanet* is very flexible, since membership in a single VLAN can be defined with any combination of port grouping, MAC-layer grouping, and network-layer grouping. A station can be member of multiple VLANs, and there is no limit to

the number of VLANs that can be defined in a *Laidanet* network. Since the switch operation and the frame format are not modified by the VLAN implementation, there is no performance degradation, and the VLAN functionality can be added, updated and removed without any change to the switches.

3.4.2. Explicit Routing An administrator can manually setup a path and add the corresponding entry in the *Controller*'s *Path Table*. The associated label will then be returned in *labelled addresses* to the stations, causing traffic to follow this path.

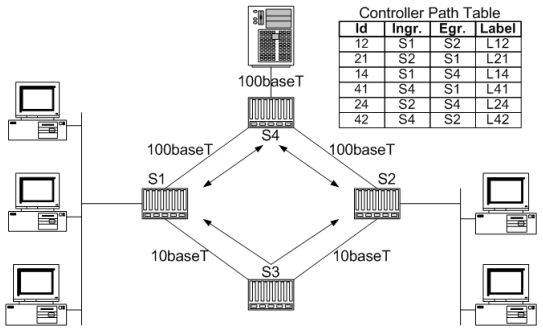


Figure 7: Explicit Routing in *Laidanet*

This technique can be used to reserve some links for particular traffic, or balance traffic between multiple paths. In figure 7 for example, the 10baseT links are used only for traffic between stations, while the 100baseT links are reserved for traffic between stations and the server.

3.4.3. Protection Against Fault The *Controller* can configure alternative paths, and setup the switches to immediately switch the traffic on those paths when a fault occurs, as depicted in figure 8.

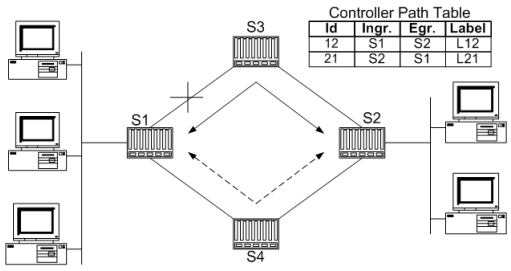


Figure 8: Protection Against Fault in *Laidanet*

A *Laidanet* network could then recover quickly from faults, and experience lower fault-related error rates.

4. Discussions

In this section, we discuss *Laidanet* benefits, and we present some results of our performance evaluations.

4.1. Advantages and Challenges

4.1.1. Why is *Laidanet* a Good Alternative for LAN Interconnection? While designing *Laidanet*, we expected that its benefits will mainly derive from two features. The use of the label switching technique, which allow to enhance the services provided, and the presence of the *Controller*, which results in the simplification of switch functionalities.

Since there are two labels in a frame, the traffic is aggregated in the switches. Only one bidirectional path is necessary to forward the traffic between any two LANs interconnected by *Laidanet* switches, regardless of the number of hosts in each LAN. The forwarding table size is then reduced, particularly in core switches. Since labels are carried in a standard Ethernet frame field, the network do not experience extra traffic load due to the use of labels. Moreover, there is no need for an adaptation layer between peripheral LANs and the core *Laidanet* network. The use of label switching technique allow the use of more advanced traffic management techniques in the network, such as explicit routing and protection against faults.

Thanks to the presence of a *Controller*, the switch functionalities are simplified. They do not implement the STP, and they run the *Backward Learning* process only for a subset of hosts: those directly connected to a local port. All the configuration is managed by the *Controller*, which then insures that the network is loop-free. Since the STP is no more used, a *Laidanet* network offers a better utilisation of its resources, since all ports on switches can be made active without any risk of loops in the network.

4.1.2. Which Challenges are Introduced by *Laidanet* The *Controller* is at a central position in a *Laidanet* network. In other to minimize the risk of fault, a distributed implementation may be envisaged, at the cost of a higher complexity in both processing cycles and traffic load.

The *Laidanet* operation depends on the address resolution protocol in use in the network. The *Controller* and the switches must be able to understand and interact with the address resolution mechanism for each L3 protocol in use in the network. Moreover, the *Laidanet* operation requires that the address resolution requests are either broadcasted or multicasted by the hosts. There is no way to interact with a host-internal address resolution mechanism, as it is the case for mul-

ticast address resolution or for hard coded address association. *Laidanet* switches must also be able to handle these particular cases, for example by recognizing the corresponding addresses and replacing them with the right labels before forwarding.

4.2. Implementation and evaluation

We implemented *Laidanet* by modifying the *bridge* module of the Linux 2.4.22 kernel. Modifications include forwarding table definitions, forwarding functions and user space utilities. We performed comparative performance measurements on both *Laidanet* and standard Ethernet bridging.

4.2.1. Measurement Environment We carried our experimentation on a set of Linux boxes, Pentium III with 500 MHz processor and 128 Mb memory. Following the standard of RFC 2544 and RFC 1242 for benchmark performance testing, we performed throughput and latency measurements for a range of different frame sizes. In order to have a more accurate view of the incidence of *Laidanet*, we ran the same tests on two linear topologies, respectively with 2 and 3 switches. We also measured CPU occupation during all tests.

We used the *netperf* utility for throughput and latency measurements, and the *sar* utility (provided under Linux in the *sysstat* package) for CPU occupation measurements.

4.2.2. Results and Comments The results are summarized in the following diagrams.

The diagrams in the figure 9 respectively represents the throughput and latency measurements. In the topology with two switches, *Laidanet* performs almost equally well as Ethernet switching, while in the topology with three switches *Laidanet* performs better. The difference is noticeable on transaction rates and throughput, both for UDP and TCP traffic. During all the tests, the average CPU occupation percentage measured with *sar* was 27.66% for Ethernet switches, and 26.61% for *Laidanet* switches.

The search algorithm implemented in the switches is very simple: input labels of 12 bits length are used as indice of the forwarding table. However, the real MAC address mut be written in the frames before they are forwarded to the destination station. *Laidanet* is then more interesting when there are at least two switches in the interconnection. *Laidanet* switch functionalities are simplified because they do not implement the STP, hence a lesser CPU occupation percentage.

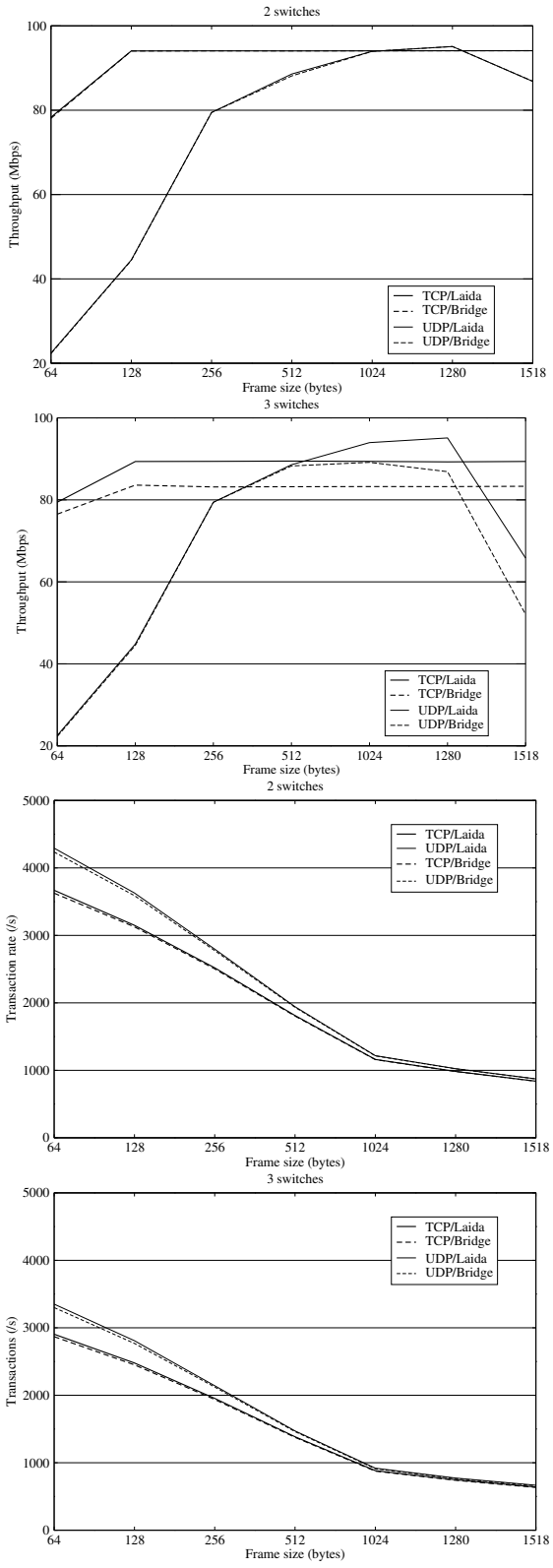


Figure 9: Throughput and transactions rate measurements

5. Conclusion

As a result to its successful evolutions, Ethernet, the most popular LAN technology in use today, has now entered various other areas, such as metro networks. Also, today's networks have scaled and applications need more bandwidth and reliability. In a switched LAN, the *Spanning Tree Protocol* is used to prevent loops and provide alternative paths. But due to its high convergence time and its restrictions on network resources utilization, it imposes many limitations to Ethernet LANs.

The combination of Ethernet and label switching have been investigated to overcome these limitations and enhance LAN services, first with ATM LAN Emulation, and more recently with the introduction of Ethernet over MPLS in metro networks. Although the proposed architectures do not fit well with the requirements for low cost and simplicity of most LANs, they proved that Ethernet switching could be further enriched with the services enabled by the label switching technique. It also appeared that in a switched network, separating the forwarding architecture and the bridging architecture was a good way to simplify both configuration and switch functionalities, while preserving backward compatibility.

Laidanet results from the combination of Ethernet, label switching, and bridging-forwarding separation. Thanks to these concepts, the network configuration is easier, the switch functionalities are simplified, and the network still provides standard Ethernet services to end stations. *Laidanet* then offers enhanced configuration capabilities (explicit routing, protection against faults) and better performances than standard switched Ethernet LANs.

However, the *Laidanet* design introduces new challenges in switched LAN. First, the *Controller* is located at a central position. To secure the network, a distributed implementation must be envisaged, so the network can still operate correctly if a single instance fails. Then, the *Controller* and the switches should be able to understand and interact with the address resolution mechanism for each L3 protocol in use in the network. Otherwise, an alternative mechanism must be provided to map MAC addresses to the right labels.

References

[1] M. Ali, G. Chiruvolu, and A. Ge. Traffic Engineering in Metro Ethernet. *IEEE Network - Magazine of Global Internetworking*, 19(2):10–17, 2005.

[2] G. Chiruvolu and al. Issues and Approaches on Extending Ethernet Beyond LANs. *IEEE Communications Magazine*, 20(4):80–86, Mar. 2004.

[3] N. Feamster and al. The Case for Separating Routing from Routers. In *Proc. SIGCOMM'04 Workshops*, Portland, Oregon, USA, Sept. 2004.

[4] IEEE 802.1Q. IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks. Std, IEEE, 2003.

[5] IEEE 802.3. IEEE Standard for Information Technology – Common Specifications – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Std, IEEE, 2002.

[6] M. Médard, S. G. Finn, R. A. Barry, and R. G. Gallager. Redundant Trees for Preplanned Recovery in Arbitrary Vertex-Redundant or Edge-Redundant Graphs. *IEEE/ACM Transactions on Networking*, 7(5):641–652, 1999.

[7] D. Passmore and J. Freeman. The Virtual LAN Technology Report. Technical report, http://www.3com.com/nsc/200374.html, 1996.

[8] V. Rajaravivarma. Virtual Local Area Network Technology and Applications. In *Proceedings of the 29th Southeasthern Symposium on System Theory(SSST)*, Cookeville, Tennessee, March 1997.

[9] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, IETF, Jan 2001.

[10] R. Seifert. *The Switch Book: The Complete Guide to LAN Switching Technology.* Wiley, 1st edition, 2000.

[11] W. Stallings. *Local and Metropolitan Area Networks*, chapter 14. Prentice hall, 5th edition, 1997.

[12] The ATM Forum. LAN Emulation Over ATM 1.0. http://www.atmforum.com/standards/approved.html, January 1995.