Emulation of GMPLS-controlled Ethernet Label Switching

Wouter Tavernier*, Dimitri Papadimitriou†, Didier Colle*, Mario Pickavet*, Piet Demeester*
*Department of Information Technology (INTEC), Ghent University – IBBT
Gaston Crommenlaan 8, 9050 Gent, Belgium
E-mail: {wouter.tavernier, didier.colle, mario.pickavet, piet.demeester}@intec.ugent.be
†Alcatel-Lucent Bell, Antwerp, Belgium
E-mail: dimitri.papadimitriou@alcatel-lucent.be

Abstract— **Last decades bridged Ethernet has been the de facto standard layer 2 technology for LAN environments. This is mainly because it is easy to deploy, it has low cost and is relatively robust. These attractive properties together with highly increasing PHY speeds result in several initiatives of research projects and standardization bodies to extend Ethernet such as to use the technology in carrier environments. However these Carrier Ethernet solutions are difficult to test with given simulation tools. This paper presents an emulation platform based on open-source software that allows to evaluate a Carrier Ethernet variant called Ethernet Label Switching (ELS). The platform allows various topology setups and has the possibility to perform time and performance benchmarking and traffic generation using standard Linux software. We will present the architecture of the emulation framework and show promising results in base setups.**

I. Introduction

For decades Ethernet is dominating the LAN environment. Ethernet bridging has become a synonym for a cheap, plug-and-play and highly compatible network technology. The high Ethernet usage in companies together with globalization trends results in an ever increasing demand to providers for Ethernet services enabling to interconnect several branches of companies (see VPWS and VPLS services in next sections). A market research survey of Infonetics over 29 service providers shows that carriers report that there is an increase of 90 to 100 percent of Ethernet traffic (see [1]).

This shift towards more packet-oriented services has also its consequences on the transport technology that is being envisioned by operators. Why would they still use more expensive circuit-based optical equipment, involving several conversion layers, e.g., GFP, VCAT, etc., if the majority of services is becoming more and more packet-based (elastic and streaming traffic). This reasoning will become even more striking, given the highly increasing Ethernet PHY speeds, going from 10 Gbps towards 40 and 100 Gbps. Therefore using Ethernet directly as a transport technology in access-, (metro-)aggregation or even core networks becomes more and more attractive.

However, base principles of Ethernet bridging of learning and flooding (see section II) within a restricted virtual tree topology is far too restrictive for use as a transport technology. This resulted in a spectrum of new Ethernet technology designs that were developedby IEEE, ITU-T and IETF. Recent efforts of these standardization organizations intend to use GMPLS to control Ethernet. However, they all assume a different interpretation of the existing Ethernet forwarding plane as will become clear in the next sections. The following section will shortly discuss the shortcomings of bridged Ethernet for use in carrier environments and will give an overview of resulting efforts to overcome these, focusing on Ethernet Label Switching (ELS) in the rest of the paper.

Whereas there is a plethora of environments to simulate and emulate Ethernet bridging, yet this is not the case for these new Ethernet technologies with several changes to Ethernet control and forwarding plane. However it is highly desirable to enable time and performance benchmarking and evaluate scalability of what is so fashionably called Carrier Ethernet. Section III will give a short overview of available simulation techniques, their respective shortcomings and bring up the lack of evaluation environments in support of Carrier Ethernet. In Section IV, an emulation environment based on extended Click Modular Router and Dragon open-source software will be presented to overcome the given restrictions and enabling benchmarking of ELS. The last part of the paper will give time and performance benchmarking results that were obtained in the constructed emulation platform such as to evaluate the ELS technology. Finally, the paper concludes with future work and some summarizing remarks.

II. Evolving Ethernet

A. Bridged (VLAN) Ethernet

Ethernet bridges and switches do not require configuration of their forwarding tables or any control protocol to enable network operation. This plug-and-play character mainly results from the fact that bridges use MAC address[1] learning, flooding, and learned MAC forwarding. This means that initially MAC bridges have empty forwarding tables and broadcast any

[1] A MAC address is a 48-bit address used to identify and forward towards interfaces of Ethernet networks. In contrary to the IP address, the MAC address is non-hierarchic, meaning that a set of MAC addresses in a same network segment is not structurally related (e.g. no prefix as in IP addresses).

frames to all non-incoming ports (split-horizon flooding).

802.1d Field name	Bytes
[illegible]	6
[illegible]	6
…ERTYPE	2
…YLOAD	42-1500

802.1q Field name	Bytes
MAC DA	6
MAC SA	6
ETHERTYPE	2
VID+x	2
ENCAPETHERTYPE	2
PAYLOAD	42-1500

802.1ad Field name	Bytes
MAC DA	6
MAC SA	6
SETHERTYPE	2
SVID+x	2
CETHERTYPE	2
CVID+x	2
ENCAPETHERTYPE	2
PAYLOAD	42-1500

IEEE 802.1ah Field name	Bytes
BACKBONE DA	6
BACKBONE SA	6
BACKBONE ETHERTYPE	2
BVID+x	2
I-SID	3
CUSTOMER DA	6
CUSTOMER SA	6
SETHERTYPE	2
SVID+x	2
CETHERTYPE	2
CVID+x	2
PAYLOAD	42-1500

Figure 1 – Hierarchy of Ethernet frame headers in the IEEE standards

Meanwhile a forwarding entry is created by watching the relationship of the incoming interface and the source MAC address of the incoming frame in each node (learning). Once this relationship is stored in the forwarding table, a new incoming frame destined to a learned MAC address will only be forwarded to the learned interface. This forwarding mechanism works on a local, independent per-node basis, meaning that no state is stored in nodes concerning end-to-end connectivity (connectionless forwarding). Because Ethernet does not contain a TTL field, flooding frames (for those frames for which no forwarding entry can be found in the forwarding table) can result into frames being endlessly looped through the bridged network, and so congesting the network. Therefore a control protocol, called the (Rapid) Spanning Tree Protocol (RSTP, [2]) is used so as to restrict the physical topology into a logical tree topology such that it doesn't contain cycles.

To enable logical separation within a LAN network, the concept of the Virtual LAN was created. VLANs allow segmentation of a broadcast domain (associated to the physical topology) to multiple logical broadcast domains (each associated to a given VLAN) such as to restrict and separate the flooding domain and learning scope of MAC frames. The VLAN to which a frame belongs is identified by an additional 12-bit C-tag in the Ethernet frame header (see Figure 1[2]). More details can be found in the related IEEE standard 802.1Q.

B. *Carrier Ethernet*

Whereas the protocol as sketched in the previous subsection works perfectly well for local environments (such as campus and enterprise networks), it lacks features which are desired in provider networks:

- Isolation of several traffic streams is not possible because all traffic is handled in the same way
- Scalability of bridged Ethernet networks is limited because forwarding entries cannot be grouped, every MAC address needs to be learned individually because the MAC address space is non-hierarchic (in contrary to the IP address space)
- Not all network paths allowed by the physical topology can be taken because of the RSTP-induced restriction (some ports are in blocking state to prevent loops).
- In case of network failures, reconvergence towards a new network state can become very slow as a result of the RSTP protocol.

1) Connectionless (CL) Ethernet

A multitude of extensions have been developed to overcome the given limitations. A first category of solutions borrows from the main nature of Ethernet bridging, being a connectionless forwarding protocol. The given extensions have lead to several extensions to the Ethernet frame header format to be used, these are shown in Figure 1.

The Multiple Spanning Tree standard (IEEE standard 802.1S) allows to configure a different tree topology per VLAN. In turn, this capability allows to make better use of the physical topology by restricting them into complementary logical trees.

The Q-in-Q standard (IEEE standard 802.1AD Provider Bridging) allows two levels of VLANs: Customer-VLANs and Service-VLANs, which can be nested thanks to an additional frame header field of 12-bit, being the placeholder for the Service-VLAN.

Although both extensions allow for a reasonable degree of traffic stream isolation in combination with traffic relative prioritization using IEEE 802.1p priority codepoints, they do not solve the issue of all network nodes having to learn all MAC addresses (thus needing one entry per address) in the network, leading to the bad scaling behavior of the forwarding tables as shown more in detail in [3].

Real scalability improvements are met with the MAC-in-MAC standard , defined in IEEE 802.1AH Provider Backbone Bridging PBB. This allows to separate the provider MAC address space from the customer MAC address space using additional fields in the extended Ethernet frame header (see Figure 1).. Mapping customer MAC addresses to Provider Backbone MAC addresses at the edge of the provider backbone network solves the forwarding table scalability, as no customer MAC addresses need to be learned in the core of the Provider Backbone Network. In addition, the standard adds, as part of the adaptation of the customer MAC frame (performed at the network edges), a service tag (the I-tag) that allows for

[2] "x" in the figure stands for an additional 4 bits containing the user priority and the TCI (see related IEEE standards).

additional traffic stream isolation, for up to 2^{24} services streams.

	Connectionless Ethernet	Connection-oriented Ethernet
Classic Ethernet	IEEE 802.1D MAC Bridge - MAC address-base forwarding - Learning & Flooding - RSTP-constrained topology IEEE 802.1Q VLAN Bridge - Restrict broadcast by VID	
Carrier Ethernet	→ Evolved forwarding plane with more scalability, but maintain RSTP IEEE 802.1AD Provider Bridge - Separation C-VID vs. S-VID (Q-in-Q) IEEE 802.1AH Provider Backbone Bridge - MAC-in-MAC tunneling	→ Evolved control (eg. GMPLS) and forwarding plane (eg. label switching) Ethernet Label Switching - Link local S-VID label IEEE 802.1Qay Provider Backbone Bridge-TE - Domain-wide B-DA + B-VID label

Figure 2 - Overview of Ethernet technologies

2) Connection-oriented (CO) Ethernet

Whereas the given connectionless extensions mainly solve the first two or three limitations, they still inherit on the characteristics of the RSTP-protocol, resulting in relatively bad network usage and bad re-convergence behavior (see [4]). Therefore several protocols have been developed which depart from (by disabling) the learning, the flooding behavior and the distance-vector protocol RSTP.

Routing freedom can now be obtained by maintaining state per logical data path (connection-oriented behavior) and control protocols to discover the topology and control protocols to explicitly signal logical connections across the Ethernet network. These design objectives typically result into the re-use of advanced control protocol suites such as GMPLS (RFC 3945) making use of link-state routing protocols such as OSPF(-TE) and RSVP(-TE) for signaling (i.e. for data path provisioning).

One of the technology proposals resulting from these objectives is PBB - Traffic Engineering (PBB-TE, [5]) which builds further PBB logic, using its frame header and forwarding behavior. PBB-TE can create logical connections and forward frames based on a combination of the B-VID (Backbone Vlan-ID) and the B-DA (Backbone Destination Address). These fields are invariant along its path towards the destination (domain-wide unique label). Because in this paper we focus on Ethernet Label Switching, being the CO technology to which we dedicated the following subsection, we won't further go into detail of PBB-TE, for this we refer to [5].

C. *Ethernet Label Switching (ELS)*

ELS is a CO-like scheme that intends to use GMPLS as its control suite with OSPF-TE for routing and RSVP-TE for signaling logical data paths over an Ethernet Network. ELS uses the Provider Bridges (802.1ad) standard to perform label switching in a similar way as is done in MPLS (RFC 3031). It encodes the label in the S-VID tag field of the related frame header (see Figure 1). The Ethernet S-VID label space has link local scope and local significance: thus providing 4096 (12 bits) values per interface and allowing intermediate ELS switches to translate the S-VID value resulting logically into a label swapping operation as it is the case in MPLS networks.

The logical data paths established using ELS are called Ethernet label switched paths (E-LSP). Intermediate nodes are called Ethernet Label Switching Router (E-LSR). Ingress/egress ELSR where a LSP starts and ends, provide Ethernet Label Edge router (E-LER) functionality. Figure 3 describes the label operations along an Ethernet LSP.

When a native Ethernet frame arrives to the ingress LSR, its E-LER function based on the information of the frame header, pushes the corresponding label (i.e. adding an S-TAG with the appropriate S-VID value). Then, the Ethernet VLAN-labeled frame is forwarded along the Ethernet LSP. For each E-LSR, the label is swapped (i.e. that the incoming S-VID is translated into an outgoing S-VID as defined in IEEE 802.1ad). When the frame reaches the egress LSR, its E-LER function pops the label (the S-TAG and so the S-VID are removed). Finally, the frame is sent to its destination as a native Ethernet frame.

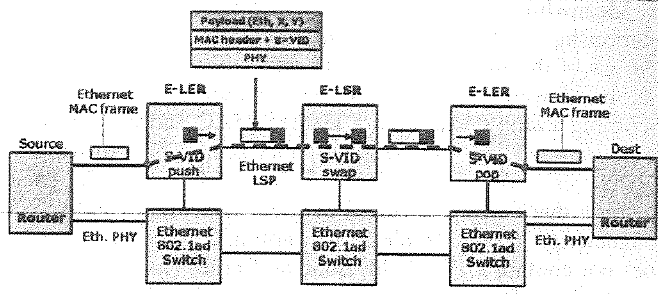

Figure 3 - Forwarding in an ELS network

It is important to underline that ELS maintains a control state per logical data path but keeps the forwarding paradigm of existing Ethernet switches unchanged except for the fact that forwarding entries are defined per port. In a sense, signaling is used to restrict the incoming and outgoing S-VID per port. The rest of the forwarding process is as per IEEE 802.1ad.

III. CARRIER ETHERNET EMULATION AND SIMULATION

A. *Challenges*

Evaluating (Carrier) Ethernet technologies is not an obvious task. Several reasons can be found for this, but the most prominent one is that except for Ethernet bridging and VLAN bridging no simulation environment supports out-of-the-box the discussed Ethernet technology enhancements (see overview in Figure 4). This is not surprising, given the fact that these Ethernet flavors are very recent.

Evaluating Carrier Ethernet technologies also sets other requirements then bridged Ethernet: time and resource performance requirements are more severe, scalability needs to be a lot higher and manageability is more important.

B. *Network simulation*

Network simulation is a technique where the properties of an existing, planned and/or non-ideal network are modeled in a software environment in order to assess performance, predict the impact of change, or otherwise optimize technology decision-making. Simulation thus involves some level of abstraction such as to calculate the interaction between the different network entities (hosts/routers, data links, packets,

...sibly using mathematical formulas. Typically ...ons are executed on a single computation system.

	NS2	Omnet++	Click	Linux
Type	simulati on	simulati on	emulati on	emulati on
Ethernet support				
Bridging (802.1D)	x	x	x	x
VLAN Bridging (802.1Q)	x		O	x
RSTP (802.1W)			O	x
MSTP (802.1S)				
PB (802.1AD)			O	
PBB (802.1AH)				
GMPLS support				
OSPF-TE	x			x (Dragon)
RSVP-TE	x	x		x (Dragon)

Figure 4 - Ethernet components in simulators and emulators (x=available, o=own development)

Evaluating scalability or deducing performance trends of technologies is a task for which simulation is well-suited. Most network simulators use discrete event simulation, in which a list of pending "events" is stored, and those events are processed in order, with some events triggering future events -- such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node.

A direct constraint of a network simulator is the dependency on the accuracy of the model that is used. Because there is not necessarily a one-to-one mapping between resources in the simulation and resources of an actual environment, simulation environments are often better for detecting certain trends than for having an accurate and sensitive platform for measuring time and system performance. Reference [tcp study] illustrates for example that simulated models typically do not account for low level entities such as cpu type, buses, network devices, etc. and that this can result into differences for example in modelled cpu load.

The most popular free simulation tools are NS2 ([6]) and Omnet++ ([7]). Figure 4 illustrates the carrier Ethernet functionality that they support. In [8] a Carrier Ethernet simulation study has been done using the TOTEM-simulator ([9]) which was extended for modelling GELS and the BridgeSim ([10]) simulator for modelling RSTP behavior. This software was used for characterizing LSP acceptance rates, bandwidth placement, link utilization and convergence time of both technology classes. The tools in the study were an excellent tool to acknowledge the trends of GELS being more efficient in most of these areas. However the fact that assumptions were made for parameters such as signaling, reservation or switching delay, and the way they are linked, indicates that the resulting accuracy of time convergence numbers are relative and depending on these. For an accurate representation of measures which are also depending on complex processes at different levels, such as cpu load, packet loss, delay or other time related numbers, more sophisticated tools are needed.

C. *Network emulation*

Network emulation is a technique where the properties of an existing, planned and/or non-ideal network are imitated in order to assess performance, predict the impact of change, or optimize technology decision-making. As emulation actually mimics the technology under benchmarking, one network device part is typically imitated by a computation system running custom software, e.g., a server blade or PC. Therefore, to emulate a whole network setup, typically a set of server blades or PCs is needed.

Emulation needs more detailed development work than modeling in simulation. The largest gain of emulation regarding simulation is that various performance measures can be made more representative and detailed as its sensitivity is close to the intended end design, and interoperability can be accurately tested. At the other hand, scalability tests require a large number of executing nodes.

Linux is a platform which can be used relatively easy for network emulation. Because of its open-source nature, the built-in protocol stack can be adapted such as to imitate specific router- or switch-alike behavior. The Click Modular Router ([11]) is a software package which builds on this opportunity by putting a set of frequently used packet processing software components at the disposal of the network engineer developing emulation equipment. Built-in components in support for Ethernet emulation are also shown in Figure 4.

IV. EMULATION ARCHITECTURE

A. *Emulation network architecture*

For benchmarking ELS technology, an execution environment was set up using the Click Modular Router to emulate the forwarding plane ([11]), and Dragon VLSR GMPLS software to emulates the control plane ([12]) (Figure 5). Both were modified such as to allow ELS forwarding and control as described in earlier sections.

UNIVERSITEIT GENT

Keio University

CALAMVS GLADIO FORTIOR 1858

3rd Gent University and Keio University Global COE Workshop

Proceedings

Monday, 23rd February 2009

At Gent University