Protecting Experimental Data Using Custom Designed Firewall Hardware

M. Grossi[1,3], F. Alfonsi[1,2], M. Prandini[4], A. Gabrielli[1,2]

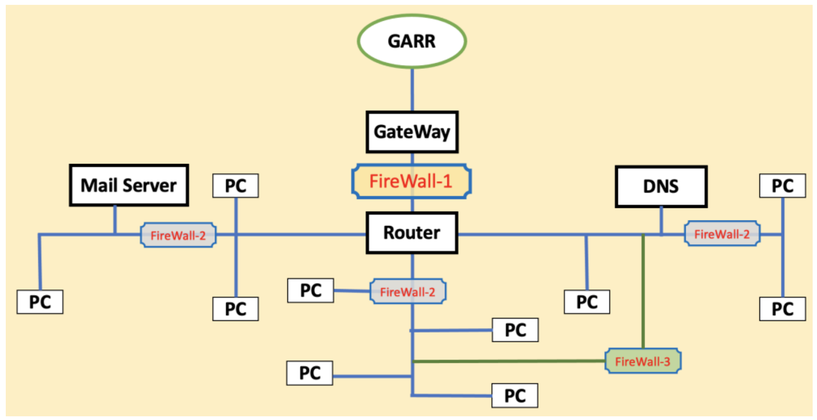
[1]Physics and Astronomy Department, University of Bologna, [2]INFN Bologna, [3]INFN-CNAF Bologna, [4]Computer Science and Engineering Department, University of Bologna

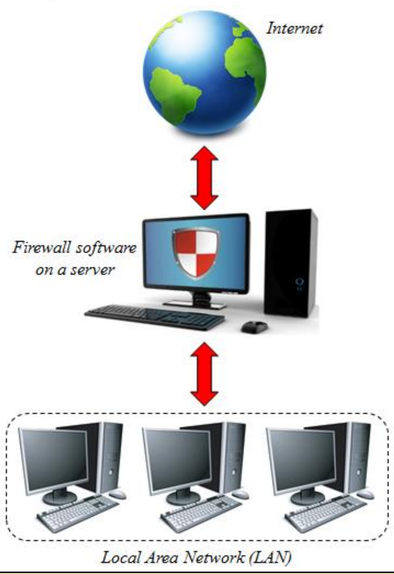
(marco.grossi8@unibo.it)

(1) Importance of data protection in science

Cybersecurity in science plays a crucial role in protecting sensitive data, research findings, intellectual property, and infrastructure within the scientific community. As technology continues to advance and more scientific processes and data become digitized, the need for **robust cybersecurity measures becomes increasingly important**.

In recent years **malicious activities** conducted over digital networks or computer systems with the intent of **causing damage, stealing information, or gaining unauthorized access** have grown many fold.





Protecting scientific data with a **firewall** is an essential aspect of **ensuring data security and preventing unauthorized access** to sensitive information. A **firewall** acts as a **barrier** between a trusted internal network and untrusted external networks, such as the internet. **Firewalls can be used in different places within a network** and the choice of **firewall** placement depends on the network architecture, security requirements, and the level of control needed for the network traffic.

Firewalls are often implemented in a software running on a server machine. However, **when the amount of transferred data and the speed of data transfer increase over a certain threshold, these systems can lose effectiveness**. In these situations, **an hardware implementation is preferred** since this can guarantee real time operations and much higher data throughput.

Firewalls can work in two different modes:

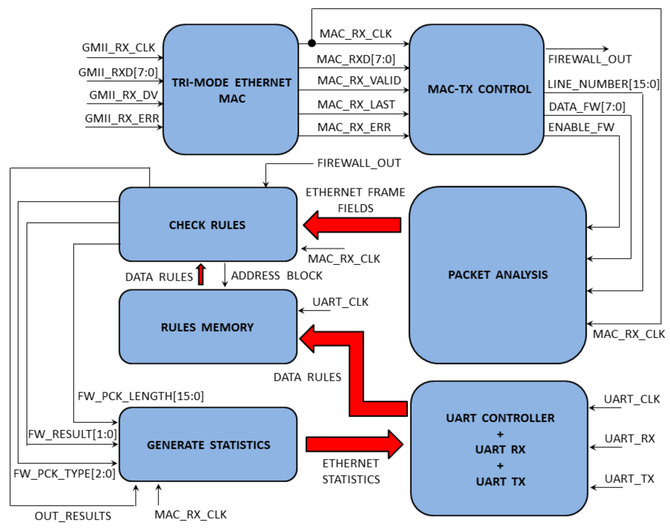
1) **Active mode**, where network **data are filtered according to a set of rules**, allowing to pass only the data that meet the rules requirements. In this case the **firewall features two different ports**, working in full duplex and when network data are presented at the input of one port, these are transferred to the output of the other port if no threats are detected or blocked, otherwise.

2) **Packet sniffer mode**, working as a **passive device** that reads all the traffic and sending alarm messages to a remote server when potential threats are detected.

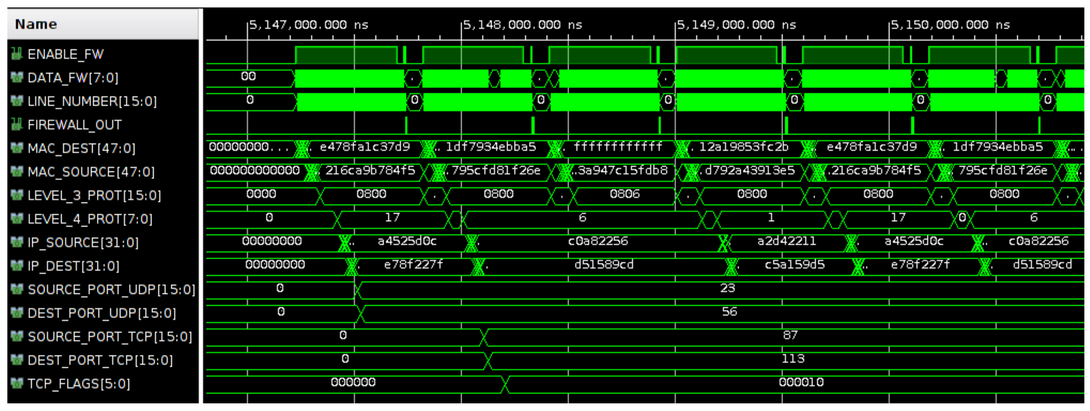
(2) Development of a firewall hardware on FPGA

A **firewall** hardware working in **packet sniffer mode** was designed using a **Xilinx KC705** development board. The system was designed in **Verilog** and works at a data transfer rate of **1 Gbit/s**. Preliminary tests were carried out to extend the system speed to **10 Gbit/s**.

- The **physical layer** of the Ethernet protocol is managed by the tri-speed Ethernet PHY chip integrated on the KC705 board (**Marvell M88E1111-BAB1C000**).
- The **data link layer** of the Ethernet protocol is managed by the **Tri-Mode Ethernet Media Access Controller** (TEMAC), an intellectual property module provided by **Xilinx**.
- The **firewall** is highly configurable, checks the Ethernet frames against a set of rules defined by the user and generates a set of statistics.

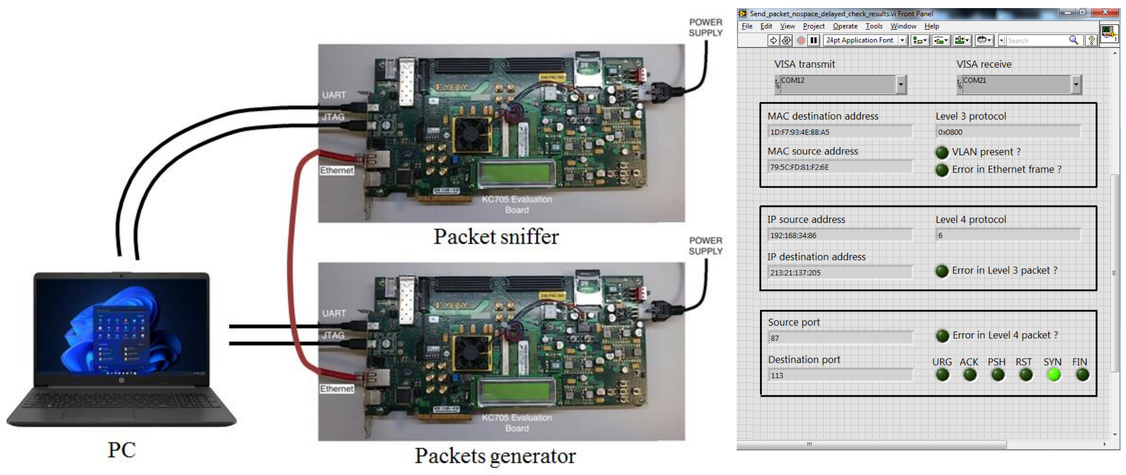


The designed **firewall** analyses the Ethernet frames and calculates important fields such as: **MAC source and destination addresses, IP source and destination addresses, source port and destination port, level 3 and 4 protocols**.



(3) Tests with controlled Ethernet frames

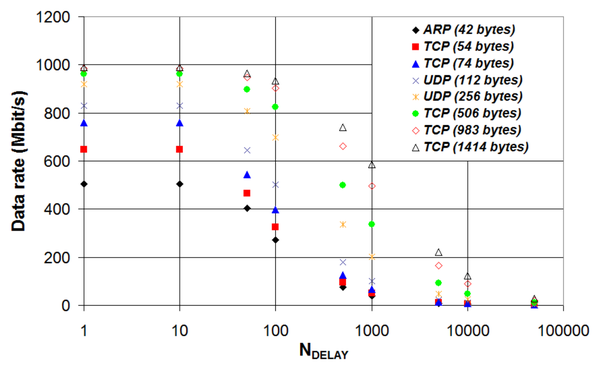
The **firewall** has been tested by generating **Ethernet frames** with a **packets generator** designed using a second **KC705** development board. The packets generator can generate Ethernet frames that are loaded in memory using the **UART port** with **inter-frames delay selectable by the user**. The generated **Ethernet frames** are fed to the **firewall** to be analysed. Both the **packets generator** and the **firewall** are controlled by **UART** using ad hoc designed **LabVIEW programs**.

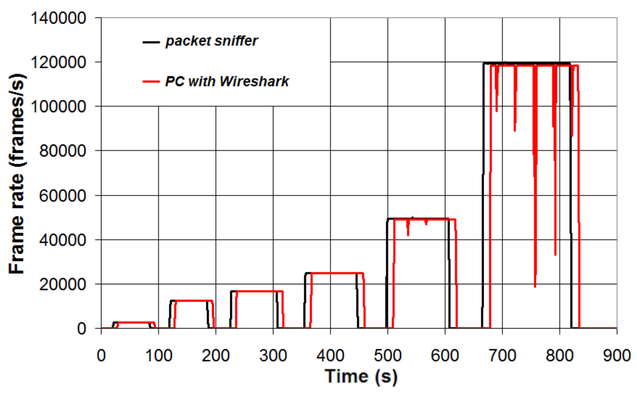


Different **Ethernet frames** of **different lengths** and **different protocols** have been tested with the **packets generator** sending data to the **firewall** Ethernet port. For each **Ethernet frame**, **different inter-frame delays** have been tested: 0, 10, 50, 100, 500, 1000, 5000, 10000 and 50000 clock cycles.

$$DATA\ RATE = 10^9 \cdot \frac{N_{ETH}}{N_{ETH} + N_{MAC} + N_{DELAY} + 12}$$

The **data rate** can not reach the **maximum value** of 1 Gbit/s, but this is the approaching **asymptotic** value as $N_{ETH} \rightarrow +\infty$. The **data rate** decreases with increasing **inter-frame delay**.

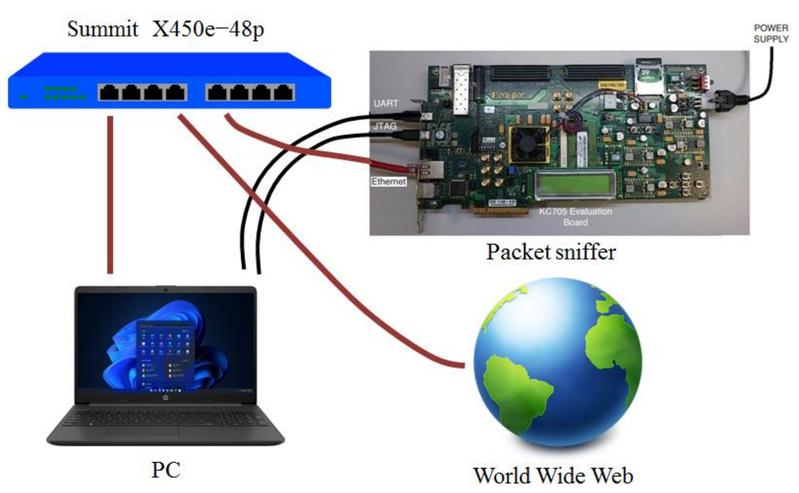




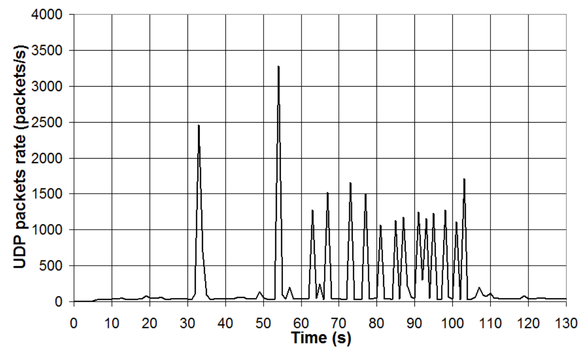
The **performance** of the proposed **firewall** have been compared with **Wireshark**, a popular packet sniffer software.
Wireshark results correlate well with the firewall data but the response of the PC running Wireshark is **delayed of about 8-11 seconds** if compared with the **firewall hardware**. Moreover, when the frames rate is higher than **48000 frames/s**, the PC running Wireshark begins to lose data and it eventually crash for frame rates higher than **100000 frames/s**.
The proposed **firewall hardware** can monitor Ethernet traffic with **data rate up to 1 Gbit/s** with **accurate results**, while this is not the case of a **PC running Wireshark** that **can not work reliably if the data rate is too high**.

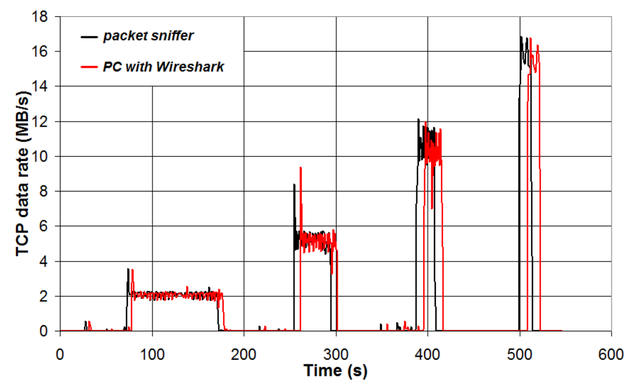
(4) Tests with real Ethernet traffic

A **switch Summit X450e-48p** by Extreme Networks has been used to **provide the same Ethernet traffic for the firewall hardware and the PC running Wireshark**. The Ethernet port of the PC is connected to port 2 of the switch and the traffic on port 2 is mirrored to port 10 where the Ethernet port of the firewall hardware is connected. A router is connected to port 6 of the switch to allow internet connection of the PC.



UDP Ethernet traffic has been generated using an on-line video streaming service. A **Youtube video** of duration of about 2 minutes was viewed on the PC using **different video resolutions (1080p, 720p and 480p)** and the corresponding UDP traffic monitored with the **firewall hardware** and **Wireshark**.
Online video streaming traffic is mainly characterized by **spiked data traffic** of Ethernet frames tagged **UDP**. The frequency of **UDP** data transfer is function of the **video resolution**.
Total **UDP** data transfer has been measured for both the **firewall hardware** and the PC running **Wireshark** and the **correlation was high**.





A **file** (Arduino IDE, file size 197 MB) has been **downloaded using the Linux command "wget"**.
The file was **downloaded four times** using **different limits for the download bandwidth: 2 MB/s, 5 MB/s, 10 MB/s and 15 MB/s**.
A **very good correlation** was found between the **firewall hardware** data and the data from **Wireshark**.
A file was also downloaded from **two different web sites**. The **IP address** of one web site is **allowed** by the **firewall rules** while the **IP address** of the other web site is **not allowed**. The **firewall hardware** correctly discriminated the two cases.

Summary

A firewall hardware implemented on FPGA has been presented. The firewall is designed using a commercial FPGA development board (KC705 by Xilinx) and supports data transfer rates up to 1 Gbit/s. The designed system analyses Ethernet frames of various types (ARP, IP, UDP, TCP, ICMP), calculates the principal frame fields (such as MAC addresses, IP addresses, source and destination ports) and evaluate potential threats of the received data based on a set of rules defined by the user. The firewall hardware has been tested under controlled conditions, generating sets of Ethernet frames using an ad hoc designed packets generator as well as under real web traffic by connecting the system to the internet. The results have shown how the designed system can reliably detect potential threats in the received data.