Hardware tap vs. PC with Wireshark:
A brief comparison

Tapping Ethernet traffic for detailed analysis requires precision and reliability that typical computers and software-based solutions like Wireshark alone cannot provide. The inherent limitations of general-purpose hardware and software make it unsuitable for scenarios where every microsecond counts, and packet loss or timing inaccuracies are unacceptable. Let us delve into the reasons why by looking at the “obvious” possibilities.

Author: Robby Gurdan

Why not using Wireshark and a fast PC?

It does sound intriguing! With a fast PC and the free software Wireshark, you have all the tools necessary to monitor, inspect, and analyse your network traffic. The new laptop might even be equipped with multi-gigabit Ethernet. Shouldn't that do the trick?

It seems that all is needed is a free port on a switch in our Network-under-Test and we are set.

Let us investigate this statement further and begin with a short introduction to what Wireshark and an ethernet switch usually do, and how they work.

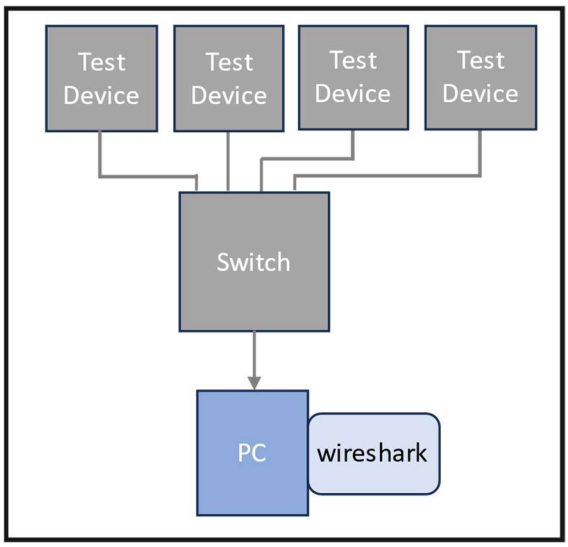


Figure 1: A PC is connected to a free port in a Network-under-Test.

What is Wireshark?

Wireshark is a well-known network protocol analyzer known for its robust features and accessibility. As a free, open-source tool, it supports a vast range of protocols and offers a graphical interface for detailed packet analysis. Wireshark's ability to decode many protocols and visualize TCP sessions provides valuable insights into network traffic. With strong community support and compatibility across multiple operating systems, Wireshark is a basic tool for network professionals seeking analysis and troubleshooting capabilities.

How does an Ethernet Switch work?

An Ethernet switch forwards packets only to ports with the appropriate MAC addresses to optimize network performance and enhance security. By using a MAC address table, the switch learns which devices are connected to each port.

When a packet arrives, the switch examines its destination MAC address and directs the packet only to the relevant port.

This selective forwarding reduces unnecessary traffic, preventing network congestion and

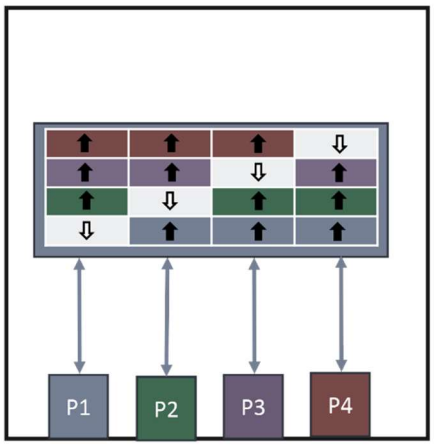


Figure 2: Every packet, that arrives at a switch port will **only** *be sent to the switch-port connected to the device wit the packets target MAC address*

collisions. It also improves security by limiting the exposure of data to only intended recipients, ensuring that sensitive information is not broadcast to all network devices.

Acknowledging all this, we have already encountered the first obstacles.

The fist obstacle to tackle, is to find a free and open port on the switch where the PC can be connected to. This additional port must be available and accessible to proceed which might already quite often NOT be the case in engineered systems.

Furthermore, using a switch and a PC with Wireshark to monitor network traffic is generally ineffective due to the switch's behaviour of forwarding packets only to the ports associated with their destination MAC addresses. This selective forwarding means that your monitoring PC will only see broadcasted traffic and traffic intended for it, missing most of the network's activity as we have seen in the explanation above.

To effectively monitor **all** traffic, one would need a managed switch with port mirroring (SPAN), which broadcasts all traffic to all ports.

So, let's move on and assume we have a managed and for us accessible (manageable) switch with both a free extra port and an appropriate mirroring feature like port mirroring or traffic mirroring.

Mirroring

Mirroring is a network feature that copies packets from a source to a destination port for analysis without affecting packet processing. Mirroring types include port, MAC, and VLAN mirroring.

Port mirroring on a switch or router sends a copy of network packets from source ports to mirror ports for monitoring and analysis.

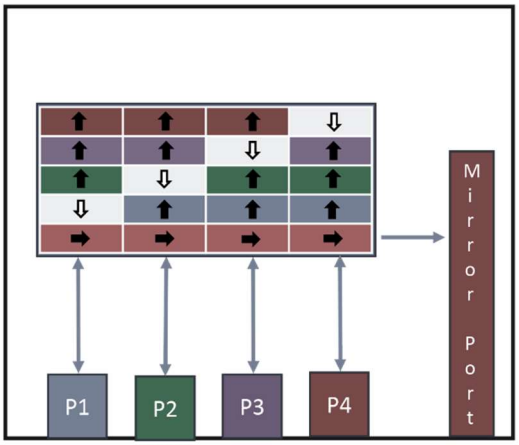
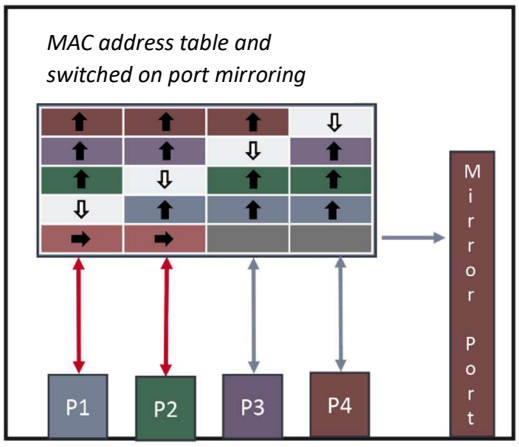


Figure 3: When port mirroring, the switch copies and forwards every packet on each selected port to another port.

Port and traffic mirroring: What is the difference?

Traffic mirroring copies specified traffic based on configuration rules to a destination port for analysis.

The source port copies traffic from a client to the destination port, which then sends it to the monitoring device. Unlike port mirroring, which copies all packets to the monitoring device, traffic mirroring only sends selected traffic.



Even if an additional port is available and capable of mirroring network traffic, it falls short in terms of precision, ensuring packet order, and providing fresh information (arrival timestamps) for transmitted packets. The forwarded packets can only be timestamped much later in software, where these timestamps accumulate delays from copying and forwarding within the switch, plus the time taken to send the data to the computer and process it before being timestamped in software. The biggest problem here is the possible data loss due to congestion, if several ports forward their data to one destination port. This data loss is not visible to the analyser, therefore it's not clear if the data was missing in the Network under Test or just dropped by the switches mirror port.

Tapping Ethernet, especially time-sensitive Ethernet traffic, demands an extremely high level of precision and reliability that typical computers and software-based solutions like Wireshark alone cannot offer. Here are the key reasons why a hardware tap is essential for this purpose:

1. Precision Timing and Timestamp Accuracy – High-Precision Timestamping:

Hardware Tap:	Computer with Wireshark:
Provides high-precision hardware time-stamping for each packet. This is crucial for analysing time-sensitive traffic where micro-second or even nanosecond accuracy is required.	Relies on the system clock of the host computer, which is less precise. Software-based timestamping introduces variability and inaccuracies due to the non-real-time nature of general-purpose operating systems.

2. Packet Loss Prevention – Dedicated Buffering:

Hardware Tap:	Computer with Wireshark:
Equipped with large, dedicated buffers specifically designed to handle high-throughput traffic without dropping packets, ensuring that all packets are captured regardless of network load.	NICs in computers have limited buffer space. High volumes of traffic can quickly overwhelm these buffers, leading to packet loss, which is unacceptable for time-sensitive data. Similarly, mirror ports on switches are susceptible to data loss. (see above)

3. Real-Time Performance – Low Latency:

Hardware Tap:	Computer with Wireshark:
Designed to operate with minimal latency, ensuring that packets are captured and processed in real-time without delay.	General-purpose operating systems introduce significant latency due to context switching, interrupt handling, and other background processes. This can delay packet capture and processing, making it unsuitable for real-time applications.

4. Resource Isolation – Independent Operation:

Hardware Tap:	Computer with Wireshark:
Operates independently of the host system, eliminating any risk of resource contention. It does not compete with other processes for CPU, memory, or I/O bandwidth.	Packet capturing and processing compete with other applications and system processes for resources, potentially leading to inconsistent performance and packet loss.

5. Network Transparency – Non-Intrusive Monitoring:

Hardware Tap:	Computer with Wireshark:
Designed to be completely transparent to the network, meaning it can capture all traffic without introducing any delay, jitter, or packet modification.	Must use NICs in promiscuous mode, which is less efficient and can still miss packets under high traffic conditions. Additionally, the act of capturing packets on a computer can introduce delays and jitter.

6. Specialized Features for Time-Sensitive Networks – Synchronization Capabilities:

Hardware Tap:	Computer with Wireshark:
Often includes features for synchronizing with external time sources (e.g., Grandmaster or GPS clocks) to ensure accurate timestamping and synchronization with other network elements.	Lacks the hardware synchronization capabilities required for precise timestamp alignment with external time sources.

7. Reliability and Stability – Dedicated Hardware:

Hardware Tap:	Computer with Wireshark:
Built specifically for continuous, reliable operation without the need for regular maintenance or rebooting. It ensures consistent performance over long periods.	General-purpose computers may require regular maintenance, updates, and reboots, which can interrupt continuous monitoring and affect reliability.

8. Conclusion:

For time-sensitive Ethernet traffic, precision, reliability, and real-time performance are paramount. Hardware taps are specifically designed to meet these requirements by providing high-precision timestamping, dedicated buffering, low latency, and independent operation. In contrast, a computer running Wireshark, while excellent for many packet capture tasks, cannot match the performance and accuracy needed for tapping time-sensitive traffic.

The inherent limitation of general - purpose hardware and software make it unsuitable for scenarios where every microsecond counts, and packet loss or timing inaccuracies are unacceptable.