Tapping Industrial Ethernet Traffic

© *Rob Hulsebos*

Version 7-September-2022

Introduction

I am often called to help solve a problem on machines and embedded systems using industrial networks: CAN, Profibus, Modbus, etc. The users often have no network knowledge, and the only reason to think that the network is at fault is an error message like "Node X not responding".

In 20 years of experience I have developed an error-finding strategy, and that always starts with finding all faults in the wiring and infrastructural components of the network. If that doesn't help, the next step is to check the configuration settings of all devices. And if that doesn't help, then the inevitable next suspect is: the application software. This is the difficult part of troubleshooting – no two applications are the same. We have to know which network messages are sent between the devices, what they contain, when they are sent, what is normal traffic and what is not, etc. In other words, we must "snoop" / "sniff" / "tap" all network traffic.

Snooping / sniffing / tapping network traffic on most 1st generation industrial networks (like Profibus, CAN) is fairly easy: buy a network-interface card that connects via USB to a laptop; and install the tapping software package that decodes the network messages into more readable form. On those industrial networks all messages are seen by anyone, and thus also by the snooping / sniffing / tapping system.

But with the increasing use of industrial Ethernet, life has become more complicated. Ethernet commonly has a star-topology, where each device has its own cable to a switch. There is no way to 'see' all the network traffic together. Connecting a tap to a cable only gives you the network traffic to/from that *single* device. Depending on the problem at hand, a different strategy is needed to catch the network messages that we need. In the chapters below, we will discuss several strategies.

Advantages
For embedded software engineers, looking at network traffic sometimes gives very insightful information about devices on a network:

- What do they send (versus: what you thought they sent)
- How quickly do they respond to incoming commands (and why is the new release slower than the old release ?)
- Which errors occur, but are silently repaired by higher-level protocols (introducing processing delays)

The tap may thus help in application development in an early stage, catching software issues before that very software is installed at customer's sites. A tap can also help to find problems in existing networks. Several examples:

Example 1
In the beginning of 2017, I was called to assist in finding a very strange problem in a machine which seemed to be related to an RFID tag reader. However, errors were reported by the device immediately *next* to the RFID reader, which went "bus-off", requiring a restart of the machine.

Since the respective vendors of both devices both claimed to have never heard of such strange problems, the customer asked me to inspect the traffic on the network, hoping that this could give a clue to the root-cause of the error.

After having inserted my tap in their network and starting the "Wireshark" protocol analyzer software on my laptop, it didn't take long before we saw massive amounts of damaged Ethernet messages, arriving every 10 minutes, and automatically disappearing again (figure 1).

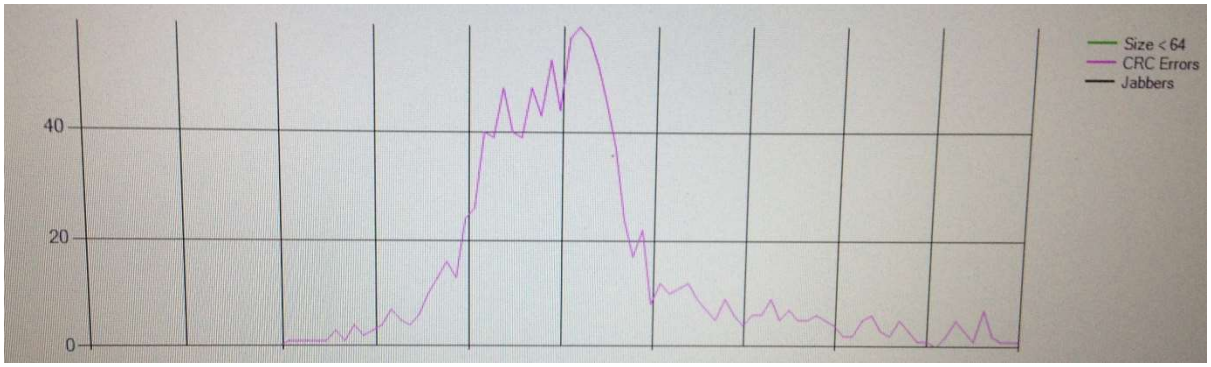

Figure 1: CRC errors caught with a network tap,
as shown by the accompanying software package (source: author)

These errors and the way they appear are not normal, and are usually caused by bad cabling, earthing problems, and/or electro-magnetic interference. After swapping devices it turned out to be a badly designed RFID device emitting the EMI and causing the network problems on another device. The vendor replaced the RFID device by a version without EMI problems.

Example 2
A customer connected his device A to my device B, and complained that *my* software was very slow. After sending a command from A to B, it took a second for the answer to be shown back on A. After having connected a protocol analyzer (Wireshark again), it was immediately clear where the delays came from: after sending the command from A to B, the answer was sent back to A within 5 milliseconds. So the network, and my device B, were not slow. It turned out that A's software needed 1 second internal processing time to show the answer.

Example 3
My (fiber-optic) internet connection at home dropped every few minutes, and came back within a minute. Tapping the network traffic between my modem and the router, it turned out that the modem thought it was connected to a dial-up line, and it was configured to disconnect the line after several minutes of inactivity. But that is not necessary on a fiber-optic connection. The modem was configured for this, but somewhere internally it still thought that it had a dial-up connection (probably a bug in its firmware). A factory reset of the modem solved the problem.

In the sections below, we will now discuss the various technologies on the market to tap network messages, and their advantages / disadvantages.

1 Use a "hub"

The use of a hub in Ethernet is one of the simplest ways to do network tapping. Just insert the hub in the data stream between two devices, and connect the laptop to the hub too. The hub's normal job is to copy each incoming message to all other connected devices, so the analyzer laptop automatically gets a copy of all that is sent.

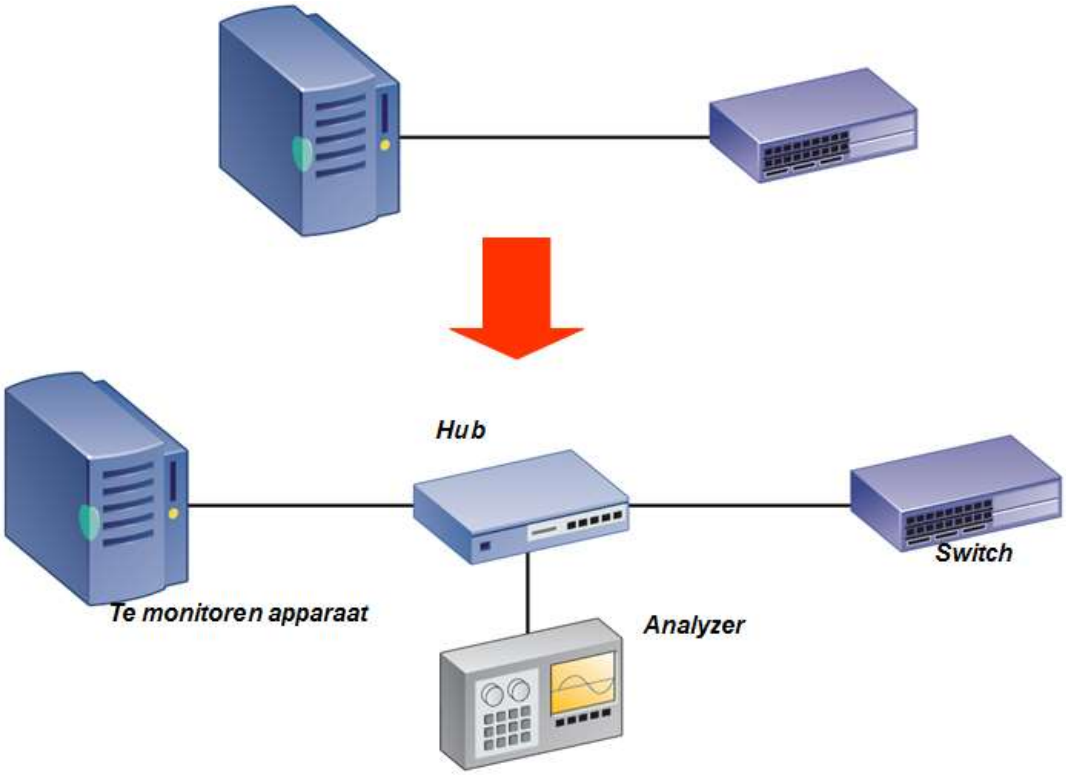

Figure 2: Using a hub to copy network traffic (source: Fluke)

Advantages:

1. Hubs have a minimal delay in forwarding messages from one port to another.
2. Multiple analyzers can be connected simultaneously to the hub – they all receive the same data, but can interpret it in their own way.
3. If you still have a hub lying around, it's the cheapest way to catch network messages.

Disadvantages:

1. Before the hub can become active, it must be inserted in the network link of the device to be monitored. This causes a short disruption in the operation of the system, so it cannot always be done when needed, but this must be planned in advance – i.e. insert the hub during a production stop.
2. Hubs are hardly sold anymore and even less used, so finding one may be difficult. Only an industrial protocol like Powerlink still works with hubs. My advice is to rescue any hub from the scrapheap where any modern network-administrator is happy to put them.
3. Hubs only work at 10 and 100 Mbit/s.

4. Hubs do not allow for the modern full-duplex mode of operation of Ethernet, only the 'half-duplex' mode. This way, devices send their message in turn. In comparison with the full-duplex mode of operation, this costs you half the bandwidth, and extra delays on the transmission time, and possibly collisions may give even more delays. It depends on the application software whether this is noticeable; in many cases it is no problem at all.

5. When a laptop or PC is connected to a hub, Windows 'thinks' it is connected to a network. It will start sending out its own network traffic, in order to start up its own TCP/IP protocol stack. This unexpected traffic may interfere with the application being sniffed!

Using a hub for network analysis purposes is a simple and cheap way of tapping simple low-speed applications. For more complex applications, one will quickly learn that using hubs have drawbacks. The next step to make in network tapping is to start using switches.

2 Use a switch, make it work like a hub

Some switches can be configured to work like a hub. For example, Hirschmann supports:

8.1.5 Gezielte Paketvermittlung ausschalten

Um die Daten aller Ports beobachten zu können, bietet Ihnen das Gerät die Möglichkeit, das Lernen der Adressen auszuschalten. Ist das Lernen der Adressen ausgeschaltet, dann überträgt das Gerät alle Daten von allen Ports an alle Ports.

☐ Wählen Sie den Dialog `Switching:global`.

Kreuzen Sie „Adressen lernen" an, um die Daten aller Ports beobachten zu können.

Figure 3: Configure a Hirschmann switch

Advantages:

It allows the use of modern switches.

Disadvantages:

1. When a laptop or PC is connected to a switch, Windows 'thinks' it is connected to a network. It will start sending out its own network traffic, in order to start up its own TCP/IP protocol stack. This unexpected traffic may interfere with the application being sniffed!

2. Duplicating all incoming traffic to all ports may influence applications that do not expect to see a lot of network traffic that they normally don't get. Although the Ethernet-electronics on the devices will filter out all network messages not intended for them, it still takes up bandwidth.

3 Use a "port-mirroring switch"

A switch which supports "port-mirroring" has a built-in feature in its internal software that allows all network messages from / to a certain port to be copied to another port – all the messages are duplicated. Which (outgoing) port they are mirrored to depends on the vendor of the switch – some allow only one port, others give you a choice.

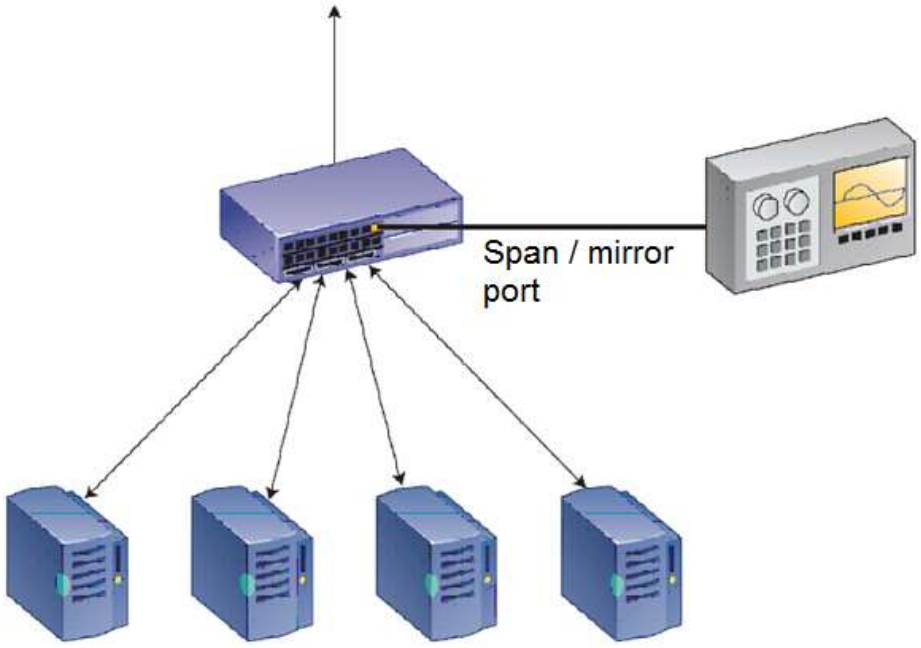

Figure 4: Using a span / mirror port (source: Fluke)

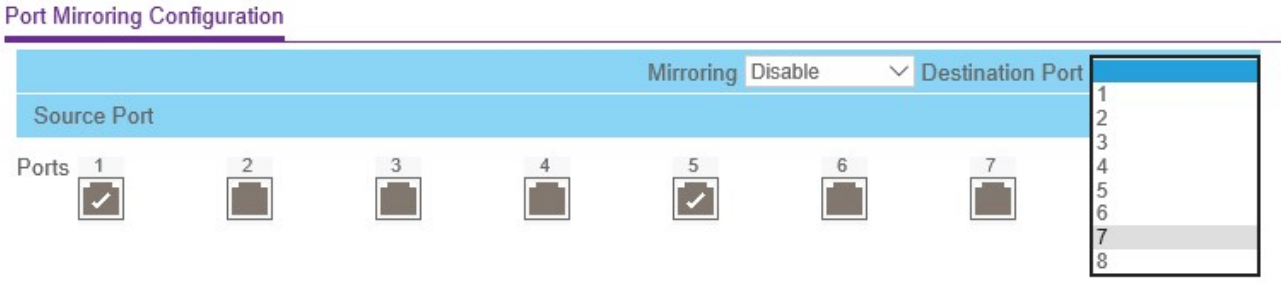

Figure 5: Example of how to configure a span / mirror port

Mirror ports are often called "SPAN" ports (Switched Port Analyzer); this is Cisco terminology.

Usually a switch only has one (outgoing) mirror port; some have two. With such a switch a message stream can be duplicated to two mirror ports, allowing two network analyzers to be connected simultaneously. An alternative solution for this is to use a "regeneration tap" (see below).

Advantages:

1. No changes in the network cabling are needed. Only the software-settings of the switch must be altered, so it knows which port(s) to mirror.

2. Port-mirroring is more and more becoming a standard feature in all managed switches. Only in the cheaper unmanaged switches there is no port-mirroring functionality.

3. There are no extra delays in handling the original messages. Especially in high-performance industrial networks, where every microsecond counts, this is of importance. There is one exception to this: if the network normally doesn't use switches (such as in Ethercat), adding a switch adds extra delay.

Disadvantages:

1. In the switch's internal processor, all incoming network messages must be filtered, and if they pass the filter, transmitted over the mirror port. But this processing takes some time, and if the switch processor is too busy (on a heavily loaded network), then the mirror port loses. This means that there is no guarantee that the mirror port will duplicate all network messages that you're interested in.

   For industrial networks, this is burdensome. We cannot detect whether any missing message has never been sent, or is never processed by the switch. It makes troubleshooting even more difficult. Only on lightly-loaded networks the switch may be able to forward all messages.

   When you use a span port to catch network traffic to be used as evidence in a court-of-law, the evidence may be rejected because you have no idea how much network traffic is missing.

2. As modern Ethernets use the full-duplex mode of operation, i.e. being able to send and receive network messages simultaneously, the bandwidth available to two devices is always twice the bitrate. For example, two 100 Mbit/s devices can generate 200 Mbit/s of traffic. This means that the mirror port must also be able to handle this, so for 100 Mbit/s network you need a Gigabit/s mirror port (and a 10 Gbit/s mirror port for two 1 Gbit/s devices). Most switches do not support this. It means that the mirror port is not able to forward all traffic (in some applications).

3. Switches normally reject for further processing any incoming Ethernet message that do not comply with the specification, i.e. a message that is too long, too short, has corrupted data, etc. Usually these are the result of (external) EMC disturbances, bad connectors, bad cabling etc. Such errors are silently repaired by protocols like TCP, so a user will not even notice what happened. But for troubleshooting purposes, the occurrence of damaged network messages is very useful information, but on a mirror port this information is lost and subsequently the network appears completely error free!

4. The mirror port must reside on the same switch as the port(s) to be monitored. Note: some switch vendors allow "remote mirror ports", for example Cisco with "RSPAN" (remote SPAN).

Using switches with port-mirroring is a good way to analyze complex systems, where it is no problem if sometimes a network message is 'lost'. But for many industrial systems, timing of network messages is of the utmost importance, and not seeing certain messages on the network analyzer makes troubleshooting very difficult. For such types of applications, the next step to make is to use a 'tap'.

4 Use a "tap"

A "breakout tap" or simply "tap" is the name for a small device which is inserted in the network link between two devices (see figure). It is transparent for the network messages sent from left to right (or vice-versa). It makes copies of all incoming messages, which are sent to the network analyzer. In this regard the tap functions more or less the same as the hub discussed in section 1.

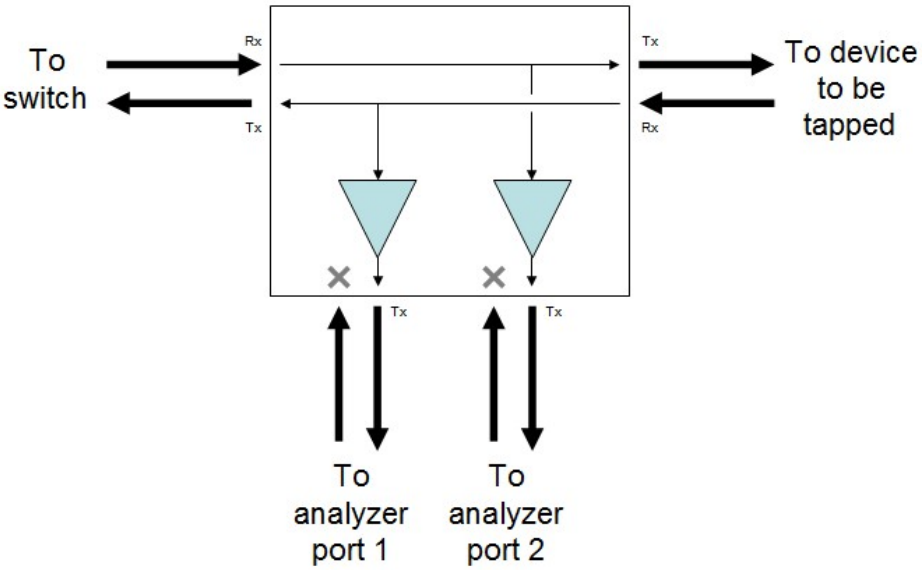

Figure 6: Dataflow in a 100 Mbit/s tap

Disadvantages:

1. In order to catch all traffic at the network analyzer, it must have two network interfaces. Most laptops have only one. Of course, a USB-controlled Ethernet interface can be added. But then you still have the problem of merging the two message streams: which messages was transmitted when? Some educated guessing might help to do this manually, but for thousands of messages this is not practical.

2. If the electronics of a tap fail, or its power-supply stops, no more communication is possible. Some suppliers solve this by allowing two power-supplies, and/or by adding a relay that automatically connects the incoming/outgoing port as soon as power is lost. Other suppliers may allow a redundant power-supply to be connected.

Examples

The cheapest tap on the market is the product of HakShop.com, selling a very simple tap which you must solder yourself (it shouldn't take more than 10 minutes). It can handle 10, 100 and 1000 Mbit/s Ethernet links, but 1000 Mbit/s are purposely downgraded to 100 Mbit/s. This makes it suitable for relatively simple applications, but what can one expect for only $10 ?

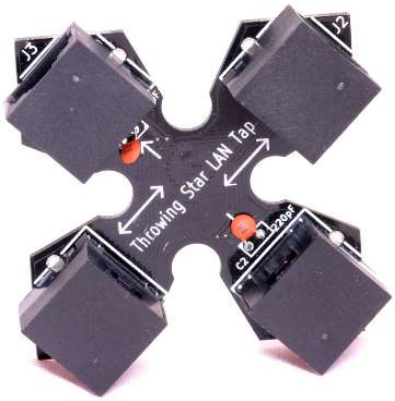

Figure 7: A simple tap, after soldering (source: HakShop)

An advantage is that it requires no external power supply. This is because all incoming electrical signals are just sent out over two ports. Ethernet doesn't expect this, as it gives extra signal degradation. But for small networks, this should pose no problem.

More expensive is the tap from Barracuda Networks:

Figure 8: The Barracuda tap (source: Barracuda)

In contrast to the Throwing Star tap the incoming Ethernet signals are electrically regenerated and retransmitted, so a power supply is needed for this tap. It functions at 10 and 100 Mbit/s only.

Another example is Fluke's "TAP100" (price about $480) or the "TAP10-100-1000" supporting more possible bitrates.

Figure 9: The Fluke tap (source: Fluke)

A tap specially made for industrial Ethernet applications is Hilscher's "netMirror". It is intended for permanent installation on a standard 35mm DIN-rail, has a 24V powersupply, and supports 10 and 100 Mbit/s. This product specifically mentions in its documentation that on the "mirror out" ports there is no circuitry for receiving data. This is important in industrial systems as no data can ever be sent *into* a network by the analysis device.

Figure 10:The Hilscher "netMirror" tap (source: Hilscher).

The German company "Helmholz" has one of the smallest industrial taps on the market:

Figure 11:The Helmholz "TAP IE100" (source: Helmholz).

5 Use an "aggregating tap"

An "aggregating tap" works identically to a tap as described in the previous section, except that it combines the two message streams into one outgoing stream. This means that the network analyzer only needs a single Ethernet port.

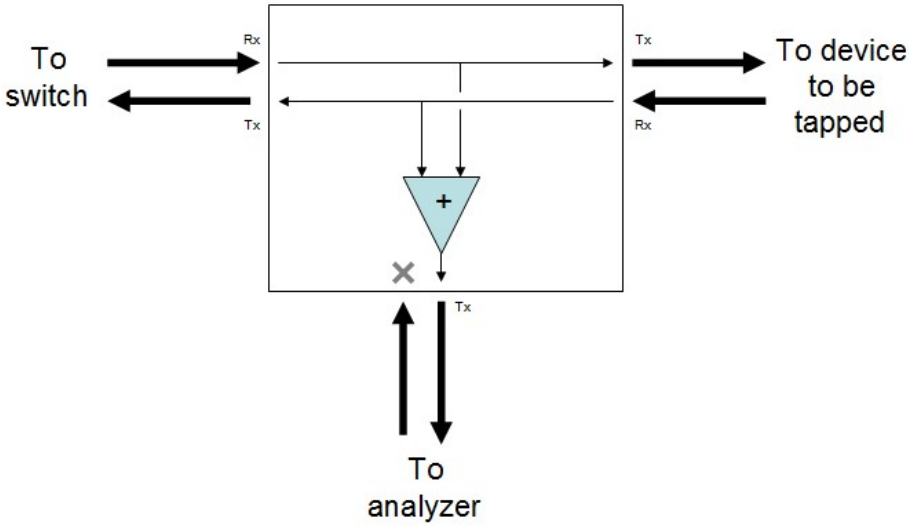

Figure 12: Dataflow in a 100 Mbit/s aggregating tap

Due to the way Ethernet works, aggregating taps have a peculiar disadvantage: the bandwidth on the outgoing port must be double that of the incoming ports. This means that two 100 Mbit/s links require an outgoing 200 Mbit/s link, but since that speed does not exist in Ethernet, it must be a 1 Gbit/s link. When tapping two Gbit/s links, the outgoing link must be a 10 Gbit/s link. This is a problem, as there are no laptops that have 10 Gbit/s Ethernet interfaces.

Examples

An example of an aggregating tap is Fluke's "ATAP100":

Figure 13: The Fluke aggregating tap (source: Fluke)

These are actually brand-labelled products from DataCom Systems:

Figure 14: The DataCom aggregating tap (source: Datacom)

Combinations of aggregating and non-aggregating taps

Some vendors have taps that have aggregating and non-aggregating functionality combined. Depending on the monitor port chosen, one or both data streams are sent out.

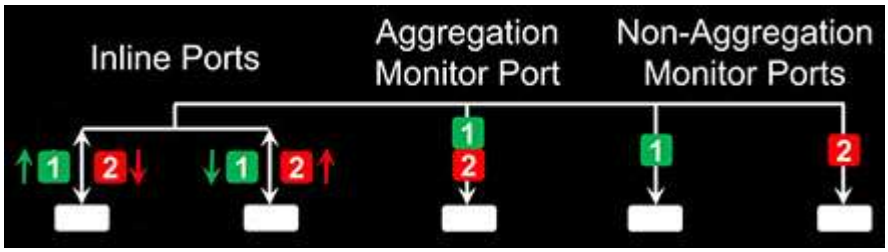

Figure 15: Two data streams combined, or not (source: DualComm)

Industrial tap

An example of a tap meant for permanent installation in an industrial application, mounting in a 35mm DIN-rail, 24V power supply, and operation in a -20..+60C environment, is the Procentec "Atlas".

Figure 16:The Procentec "Atlas" tap (source: Procentec).

6 Use an aggregating tap with USB

As mentioned in the previous chapter, the aggregated bandwidth is twice that of the tapped ports. The Ethernet solution would be to use an outgoing port with a ten times higher speed, i.e. a 10 Gbit/s link when tapping 1 Gbit/s link.

A simpler solution is possible, by using an USB3 interface (using the blue colored connectors). It supports a speed of 4.8 or 9.6 Gbit/s, sufficiently larger than 2 Gbit/s to be able to handle this.

On the PC / laptop, a special USB driver simulates an Ethernet interface on Windows. This makes that standard analysis software (i.e. Wireshark) can use the USB-tap without any further changes.

Advantages:

1. Cheaper to implement than an Ethernet 10 Gbit/s links.
2. Not dependent on a laptop or PC with a 10 Gbit/s interface, which are very scarce.
3. USB3 can supply the power for the tap's electronic circuits, so no external power-supply is necessary.

Disadvantages:

1. When the laptop connected to the tap is taken away, the tap is no longer powered; if the ingoing/outgoing ports are not automatically connected to each other (i.e. via a relay), network communication is disturbed.
2. When the laptop goes into 'sleep' mode, the power to USB devices may be switched off. This means that the network link, in which the tap is active, becomes inoperable. Depending on the vendor of the tap, a relay may be activated which connects both network ports with each other, but there may be short delay (i.e., 1 second) before this happens.
   This is only of importance for taps designed to operate at 1 Gbit/s, because of the electronics circuit in the tap. For 10 / 100 Mbit/s taps, there is no electronics inside, so these can operate without power.
3. The tap can only be connected to non-Windows analyzers if there is USB-driver software available.

Examples

Examples of USB-aggregating taps are the products of ProfiTap, i.e. the "ProfiShark" that supports datalinks with speeds up to 1 Gbit/s:

Figure 17: The ProfiShark aggregating tap for 1 Gbit/s (source: ProfiTap)

The "ProfiTap" supports speeds up to 100 Mbit/s, which is sufficient for many industrial Ethernet protocols, such as ProfiNet.

Figure 18: The ProfiTap aggregating tap for 100 Mbit/s (source: Procentec)

Another example of a USB-tap is the Hak5 product "Plunger Bug" (100 Mbit/s max). Very cheap, and you can even capture on mobile devices.

Figure 19: The Hak5 "Plunder Bug" (source: Hak5)

7 Other sorts of taps

Regeneration taps

A regeneration tap copies incoming network traffic to multiple outgoing ports (1:n). This allows more than one analyzing device to be connected in parallel, since they all receive the same network messages.

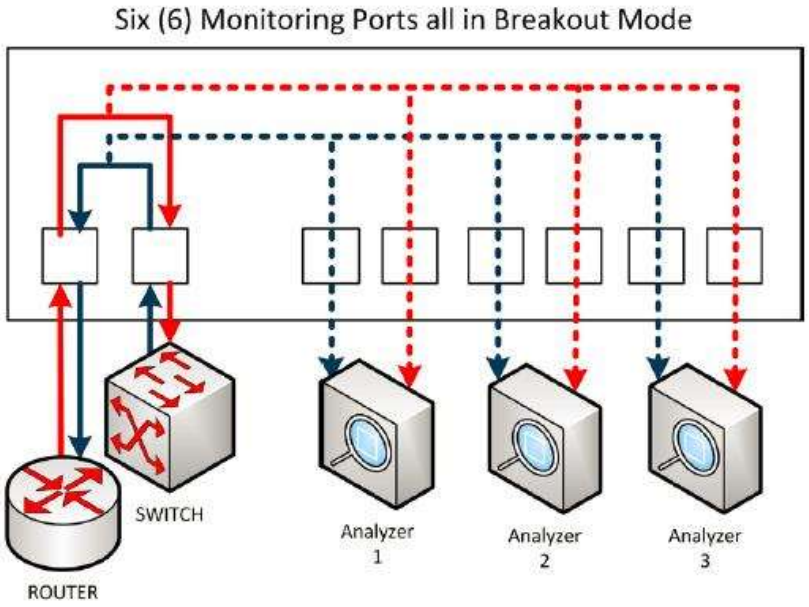

Figure 20: Dataflow in a regeneration tap (source: Garland)

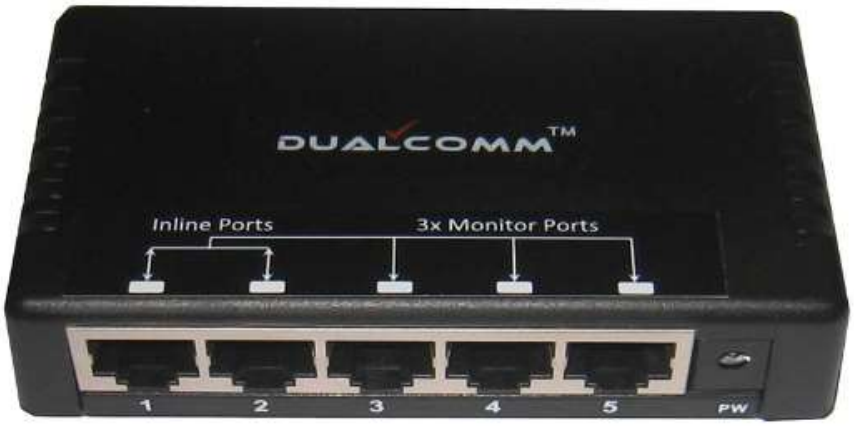

Figure 21: Example of a regeneration tap with 3 monitor ports (source: DualComm)

Multi-channel taps

A multi-channel tap allows the tapping of several Ethernet links simultaneously. All network messages are aggregated and sent out to the analyzer over one port. It is exactly the reverse of a regeneration tap.

An example of a multi-channel tap is the Beckhoff ET2000, specially made for systems running the Ethercat protocol. Since Ethercat runs at 100 Mbit/s, monitoring 4 Ethercat links requires a combined bandwidth of 800 Mbit/s, reason why the outgoing port uses a speed of 1 Gbit/s.

Figure 22: The Beckhoff ET2000 Ethercat tap (source: Beckhoff)

DIN-rail taps

All the examples on the previous pages showed taps for a service-engineer: portable, small, light-weight, separate power, easily deployable.

But for more permanent installations, and/or systems where more than one line needs to be tapped, a DIN-rail mountable tap with multiple ports is a better solution than working with many one-link taps.

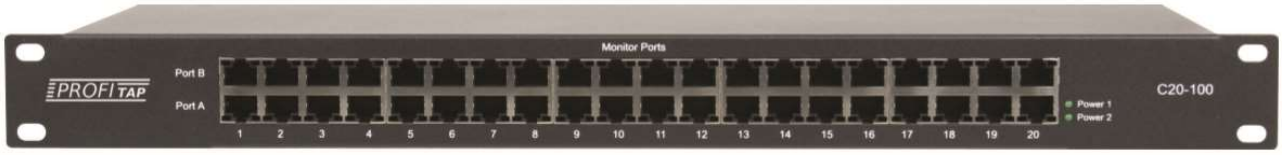

Figure 23: A DIN-rail tap (source: ProfiTap)

Fiber-optic taps

Not only (copper) RJ45 links need be tapped, but sometimes fiber-optic links as well. With a fiber-optic tap, such links can be handled as well.

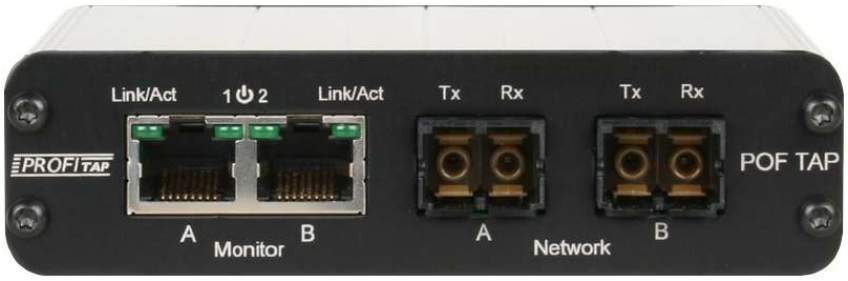

Figure 24: A tap for fiber-optic cabling (source: ProfiTap)

Bypass tap

A "bypass" tap (sometimes also called "bypass switch") functions like a normal tap, with the difference that the tapped network messages are *first* sent to an external device, like a firewall or an IPS (Intrusion Prevention System). If this device allows the network message to be passed, it is sent back to the bypass tap, which then sends it out again (figure 23).

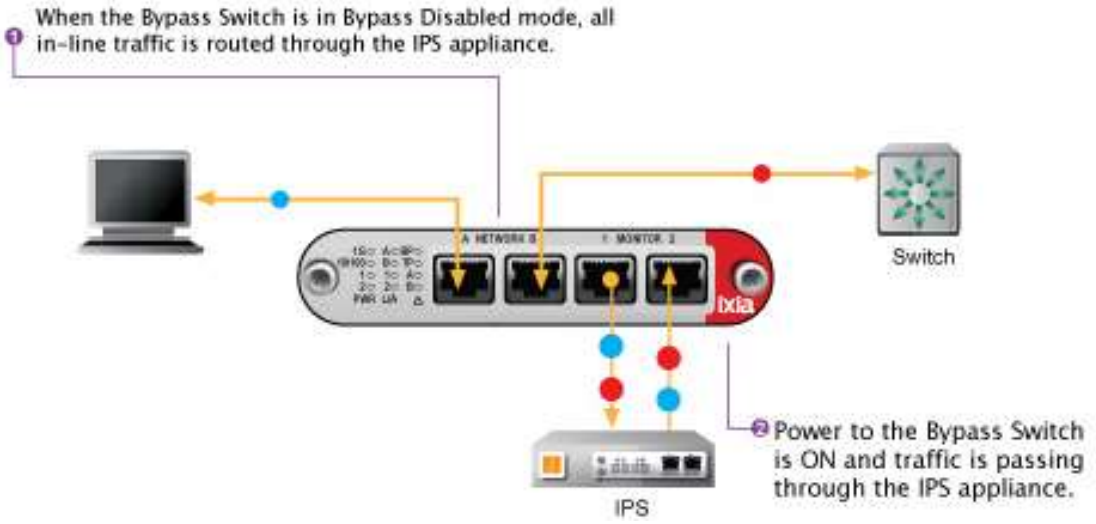

Figure 25: The normal flow of network messages through a bypass tap (source: Ixia)

It is of course possible to use a firewall or IPS without a bypass tap, but if that device fails all network traffic is blocked from coming in / going out. It is here where the bypass tap adds value: if the bypass tap detects that the firewall / IPS is no longer functioning, it no longer forwards network traffic to that device, but just sends incoming traffic immediately out again ("bypass mode", figure 24). It also does this upon a power failure.

How does the bypass tap 'know' that the firewall / IPS is operational? These devices must regularly send so-called "heartbeat" messages. As long as the tap receives these, it will forward incoming network messages. When heartbeats are no longer received, bypass mode is activated. When heartbeats are received again, i.e. following a power-on/reboot/restart of the device, bypass mode is disabled again.

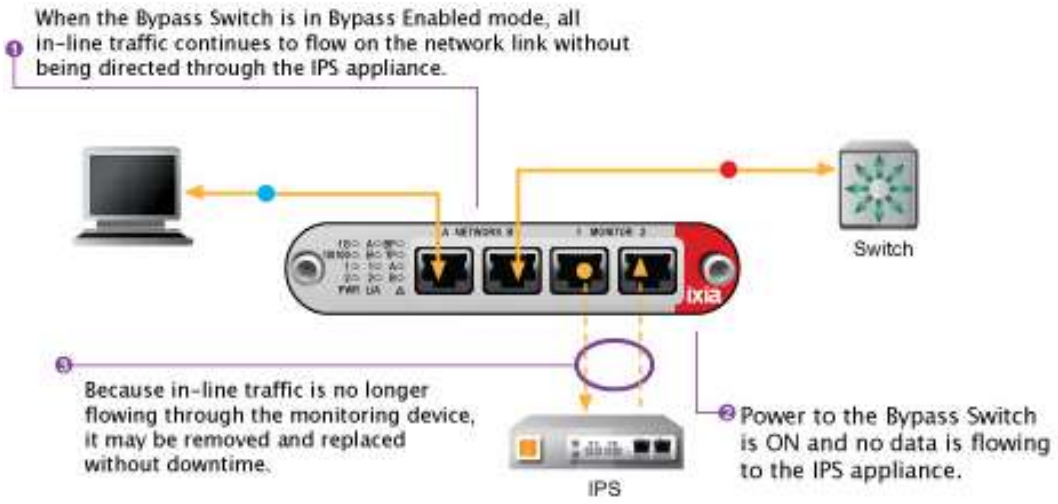

Figure 26: A bypass tap in bypass mode (source: Ixia)

For testing or maintenance purposes, bypass mode can also be controlled manually. For example, this allows a firewall / IDS to be replaced or its software updated without causing any interruption in the normal traffic flow.

Taps with protocol knowledge

All the taps discussed so far are generic – they are unaware of the protocol(s) being used on the network. On its own this is very useful, as the tap can be used in any network; but the lack of knowledge of a protocol enforces that the analysis must be done elsewhere.

But a tap can be extended to have knowledge of certain network protocol(s). An example of this is the Indu-Sol "iPNMA" which has ProfiNet knowledge and thus knows what is normal and what is abnormal on a ProfiNet. This product is meant for permanent installation in a network, directly between the PLC (Programmable Logic Controller) and the first ProfiNet device.

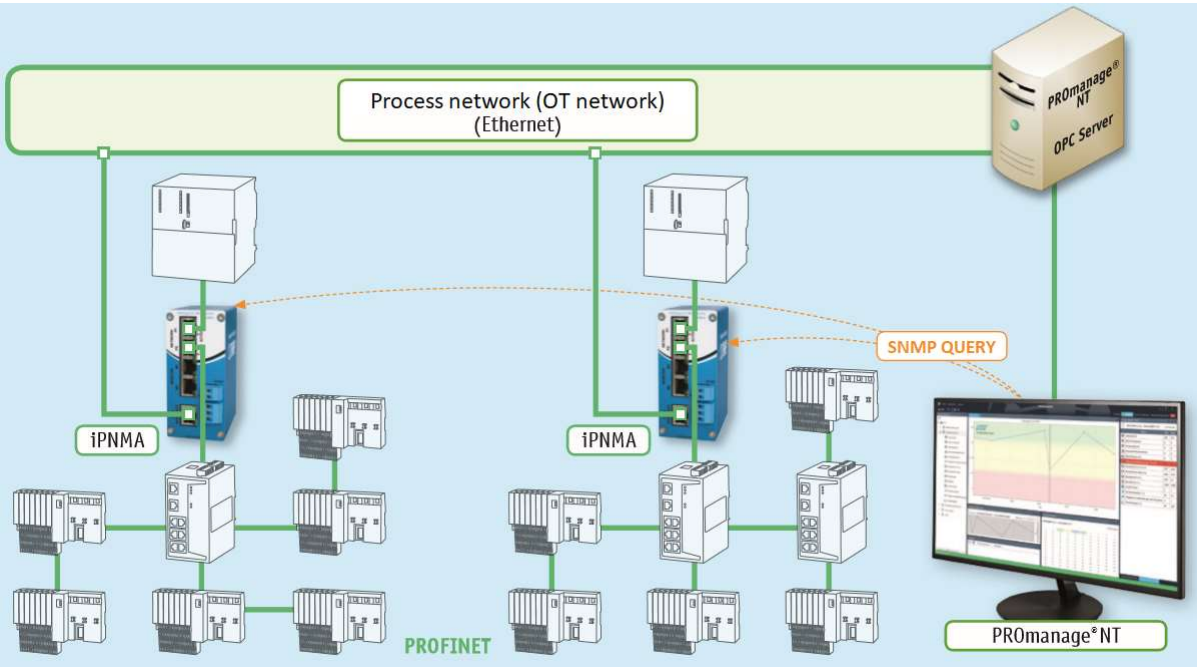

Figure 27: Two InduSol "iPNMA" taps inserted in a ProfiNet system (source: Indu-Sol).

With its ProfiNet knowledge, the iPNMA can detect unwanted changes in the protocol cycle-time, detect devices that no longer respond or respond too slow, devices that restart, too much jitter in the cycle-time, retries, damaged network messages (CRC errors), and a too high network load (%).

The 3rd and 4th port on the iPNMA can be used to connect a diagnostics laptop/PC or a dedicated analysis tool. Because the iPNMA is always present in a network, connecting diagnostics tools is always possible, and it can be done without interference on the ProfiNet to be monitored.

The diagnostic results can be sent via SNMP to a central management station (ProManage NT, also by Indu-Sol) via a 5th network port that can be connected to a higher-level process network. This port is also accessible via a web-browser.

8 Features

In the previous chapters, we have discussed taps looking only at the data flowing over an Ethernet link. But on an Ethernet link a lot more is going on:

- Handling of transmission errors
- Filtering
- Auto-negotiation
- VLAN's
- Jumbo frames
- Power-over-Ethernet
- Link failure propagation
- Timestamps

Handling of transmission errors

As on any network, messages may get corrupted while 'in transit' on the cable. This may have various reasons, such as electro-magnetic interference (EMI), bad electronics, bad cables, too long cables, etc. When a message is corrupted, it is automatically discarded as soon as it is received on a switch or network controller card. Usually this is no problem, as higher-level protocols (such as TCP) detect this and repair it.

But when there are too many corrupted messages, the network bandwidth decreases, delays occur, and applications may not receive their needed messages in time. Also, when a device detects too many corrupted messages, it may stop and go "bus off". This is a behavior implemented in many industrial network protocols.

When using a hub / switch as tap, discarding corrupted messages is standard behavior; they are not passed on to the network analyzer. When using a tap, corrupted messages are passed on to the network analyzer, who cannot do much with the data in the message, except maintaining statistics counters depending on the sort of corruption (too short message, too long message, wrong CRC, etc.)

Filtering

Because of the very high bandwidth of Ethernet (at 100 Mbit/s, even more so at 1 Gbit/s), a laptop used to store the tapped messages on can easily be overwhelmed by the enormous flow of data. Even when it is capable of storing all the data in a file on disk, processing the data can take some time.

Figure 28: A tap with filtering capabilities (source: KunBus)

Modern taps with built-in intelligence can do the filtering in the tap itself, sending only the expected messages on to the network analyzer. An example of such a tap is the "TAP Curious" made by Kunbus (Germany). Via its own webserver it can be programmed to filter incoming messages. Several filters can be programmed. Only messages that are accepted by a filter are sent on to the network analyzer[1].

Start of message acceptance

With most taps, any message they receive is passed on to the network analyzer. If a programmer is looking for a particular problem, the way of working is usually like:

1) Start the network analyzer software (start taking in network messages)
2) Start the application software
3) Wait for the problem to occur
4) Stop the network analyzer software (no more taking in of network messages)
5) Analyze the data.

The problem is that the time between 1) and 4) is sometimes quite long, and then thousand to millions of network messages remain to be analyzed. With proper filtering capabilities in the network analyzer, the number of messages to be analyzed gets smaller, but it still takes time. And hopefully you have caught the messages exposing the problem; if not, repeat the procedure!

It would be better if the tap can be started (and stopped) automatically, i.e. under control of the application software itself. The Curious tap (as mentioned in the previous section) has a digital input which can be used for this. It can be connected to a controller, a PLC, or just a normal switch.

[1] Unfortunately this product is no longer sold (7/2022).

Message detection

Sometimes a tap can be programmed to detect a particular network message, i.e. by a programmable filter. Depending on the capabilities of the tap this event can be used to stop taking in of network messages, control a digital output, sound a buzzer, etc. The handling of these messages can then be automated, instead of having you watch the screen of the analyzer laptop for the expected message to fly by. This is very useful during software development, or for catching hard-to-find problems.

Power-over-Ethernet (PoE)

When a device is disconnected from a switch with PoE and connected to a tap, it will have no power anymore. An alternative power-supply is needed for the device, or a tap with PoE functionality (PSE = Power Source Equipment).

Figure 29: A tap with PoE capabilities (source: DualComm)

Alternatively, some taps have "PoE pass-through" capability.

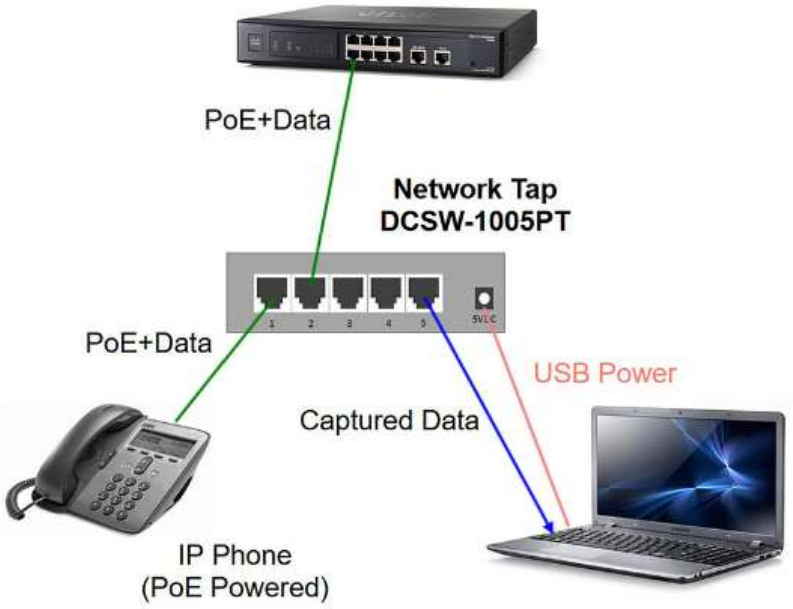

Figure 30: A tap with PoE Pass-through capability (source: DualComm)

Auto-negotiation

A tap inserted in a Gigabit link will execute the auto-negotiation handshake on both devices it is connected to. This might sometimes give unexpected results.

For example, suppose that device A supports 10/100 Mbit/s and device B supports 10/100/1000 Mbit/s. When A and B are directly connected, the outcome of the auto-negotiation handshake will be 100 Mbit/s, as this is the highest common bitrate. So, device B will work at a lower speed.

Now suppose that a gigabit tap is inserted between A and B. Both A and B will now auto-negotiate with the tap. In the direction of A, the tap will work at 100 Mbit/s, but in the direction of B the auto-negotiation will end up at 1000 Mbit/s. Now A and B will work at different speeds.

Jumbo frames

The so-called "jumbo frames" are a feature of Ethernet where a network message can have a total length of some 9000 bytes (and sometimes even larger), instead of the usual 1500. This is particularly efficient when sending lots of data, as there is less overhead and the CPU has to handle less messages. The feature is not common in industrial networks, but in case jumbo frames are used anyway: equipment that does not recognize them, assumes that the jumbo frames are erroneous an discard them.

Link failure propagation

The two devices connected to each copper Ethernet cable send "link" pulses to the other party. This feature of Ethernet allows a device to detect that there is someone else on the other end of the cable. Conversely, the *absence* of a link pulse means that there is a problem: no device at the other end, no power supply at the other end, no functioning network electronics, no cable inserted, or a damaged cable. Networks may use the absence of a link pulse to activate backup communication channels.

When a tap is inserted between two devices using redundant communication channels, it is of importance that the tap does not generate its own link pulses, but propagates the link pulses of the other two devices (as if the tap is not there). A tap that does not do this makes that the original two devices now receive the link pulses from the tap, and then a failed device is not detected and the backup communication channel is not activated.

This feature is especially of importance when a tap is continuously inserted in a network link. For a tap that is used just to detect a temporary problem it is of less or no importance.

Timestamping

Sometimes it is required to know the exact moment at which a network message was transmitted. This is not the same timestamp as shown on an ordinary network analyzer: it shows the moment the message was received on the network analyzer. This is not sufficient when monitoring multiple network links simultaneously, i.e. with an aggregating tap.

Some brands of taps have therefor a feature where a timestamp is added to a network message, to be shown on the network analyzer later. This requires special software (i.e. a device driver) on the analyzer, as the timestamp data must be recognized on the analyzer.

TCP Reset

A "TCP Reset" is a special network message according to the TCP (from TCP/IP) protocol specification. When a TCP connection is active between two devices, inserting a "TCP Reset" message will cause termination of that connection, in effect stopping all data flow between these two devices. It is a security feature on a tap, activated by an "intrusion detection system" (IDS) detecting unwanted communication patterns on an network.

As taps do not normally send messages themselves, a special type of tap is needed: with a bidirectional port.

Turnover time

Taps for Gigabit-Ethernet use special electronic circuitry in the signal path. These circuits of course need a power-supply. If this supply fails in a running network the communication between the tapped devices is no longer possible. For some applications, this is unwanted.

For this reasons, some vendors add relays that automatically close upon a power failure, and activate the bypass circuit. Data can pass the tap uninterrupted, allowing the tapped devices to continue to work.

Of importance here is: how quickly do the relays close? This so-called "turnover time" determines whether an application can continue to work normally upon a power-failure of a tap, or whether the application suffers from a too long absence of communication.

9 Cybersecurity issues

Whatever method is used to tap network traffic, new devices are connected to your network (i.e., a switch or tap, a laptop, etc.). From a cybersecurity point-of-view, this automatically means: are the new devices hackable, and can they be used to intrude on the monitored network?

Basically, the answer is: there is no way to say 'yes' or 'no'. Of course, you can rely on the assurances of the vendor, but without knowing exactly what is in the box, there is no way to be sure.

The more complex a device is, the more vulnerabilities it will have. Switches are an example of this, due to the amount of network protocols they support, embedded webserver, remote management capabilities, etc. Also, the port used for mirroring is usually also usable as a normal network port, meaning that it (physically) can also *receive* network traffic. An example of this was seen in 2021, when Siemens SCALANCE switches had a bug in them that allowed this.

Taps are much simpler, often they do not have any software in them, and the physical inability for a connected to PC/laptop to send packets *to* the monitored network makes it safer. But it depends on the vendor of the tap – for example, I once worked with a tap which allowed a connected laptop to function normally on the monitored network.

Due to increased awareness of making networks as resilient as possible to hackers and malware, there is more attention from vendors about the cybersecurity of their products.

Figure 31: PacketRaven is a tap that has protection against tampering.

The "PacketRaven" tap from Neox is a tap that (so claims this vendor) cannot be tampered with. Mechanically, it has special screws that make opening the housing more difficult (special tools are needed), and it has seals on the housing it can be detected that the tap has been opened. A 'secureboot' feature checks that the firmware has a valid signature with an authorized key. Additionally, Neox can deliver the device preconfigured to your wishes; no configuration changes are possible thereafter.

10 Summary

We have shown that there are many ways to monitor network traffic, each with their own advantages and disadvantages. For simple applications one can start with using a hub or a switch with a mirror port, but for larger applications or higher network loads taps are necessary. Taps exist in all sorts, depending on their supported features.

For every industrial network service engineer, systems developer or programmer, having good network analysis equipment is a must. A tap is a device that needs be present in every toolkit. Although they are expensive, the time saved by being able to see *exactly* what is (not) going on in an industrial Ethernet is priceless. The downtime of a production system can easily be many times larger than the price of a tap.

This article is likely not complete in describing all possible tap features. If you have any suggestions, comments or additions, please do not hesitate to contact me (email: rh[at]enodenetworks.com).