Passive Flow Monitoring of Hybrid Network Connections regarding Quality of Service Parameters for the Industrial Automation

Marco Ehrlich[1], Alexander Biendarra[1], Henning Trsek[2], Emanuel Wojtkowiak[1] and Jürgen Jasperneite[3]

[1]inIT - Institute Industrial IT, OWL University of Applied Sciences, Langenbruch 6, 32657 Lemgo
marco.ehrlich@hs-owl.de, alexander.biendarra@hs-owl.de, emanuel.wojtkowiak@stud.hs-owl.de
[2]rt-solutions.de GmbH, Oberländer Ufer 190a, 50968 Köln, trsek@rt-solutions.de
[3]Fraunhofer IOSB-INA, Application Center Industrial Automation, Langenbruch 6, 32657 Lemgo
juergen.jasperneite@iosb-ina.fraunhofer.de

Abstract: Today many hybrid (wired/wireless) communication networks with high requirements regarding latency, reliability, determinism, and security are present in the industrial domain. The corresponding required network monitoring and management becomes more and more important for all related businesses and applications. Therefore, this paper investigates the passive network monitoring approach using the flow export technique based on the Internet Protocol Flow Information Export (IPFIX) for the industry. A state of the art research will be provided, a prototype is implemented, and measurements respectively an analysis is performed in order to clarify the limits and the overall usability of IPFIX for the industrial automation domain.

1 Introduction

Today's connectivity between people, products, and machines is one of the key enablers of the Industrie 4.0 developments. The combination of several communication technologies was always and is still required in order to supply the widely spread hybrid (wired/wireless) industrial systems. There is no single technology which supports all of the requirements in the industry for e.g. critical, hard real-time applications, such as ultra-low latency ($<$1ms), high reliability (99,9999%), determinism, and advanced cyber security functionalities. The integration of the 5G telecommunication networking concepts into industry will continue with this ongoing developments [WSJ17]. Additionally, the nowadays management and monitoring functionalities of communication networks become more and more crucial for the overall success of the underlying businesses [PC15]. Network monitoring is required for various applications, such as accounting, traffic engineering, cyber security, or Quality of Service (QoS) parameter provisioning and observance [TB11, FY07]. In general, it is important to monitor the status of the networks in order to determine and control the performance and the quality of the overall communication across the hybrid systems [NEWJ17].

Various technical approaches using different methods of extracting the information from network traffics are available for this purpose. The original idea in the field of passive network monitoring was about capturing and tracing individual packets. Nevertheless, in times of high-speed connections and strong requirements on cycle times or inherited security functionalities in the industrial domain, single packet capturing results in a huge and most of the time hardly manageable amount of data which needs to be analysed later on as well. Consequently, the future requirements for network monitoring using single packet capturing are not satisfiable and are probably too expensive to purchase, implement, and maintain [HCT$^+$14]. The industrial automation networks for communication currently increase in size and performance, use hybrid communication, and are designed very dynamically. Consequently, there is more information available and configurable in the network devices resulting on the one hand in a higher effort for parametrization and configuration. But on the other hand there are many possibilities to profit from these information as well by creating new services and business models. Therefore, a resource sensitive approach is required for the industrial automation domain which is able to deliver all the required statistics about the network status.

Nowadays the mostly used network monitoring technique is flow export with IPFIX [HCT$^+$14]. The Internet Engineering Task Force (IETF) defines a flow as a set of IP packets passing an observation point in the network during a certain time interval, so that all packets belonging to a particular flow have a set of common properties [IET13]. Common properties are e.g. information from the packet's header like source and destination addresses or port numbers [HCT$^+$14, Bro11]. The result of the flow exports are so called flow records, which represent a connection between two communication partners providing a detailed view on the network traf-

fic [TB11]. Passive network monitoring solutions can be widely deployed without additional hardware, can be used in many different application domains, and provide significant data reduction compared to packet capturing [HCT+14]. Moreover, this approach scales adaptively to various network sizes in distributed architectures and is very sensitive to private information [HCT+14, TB11, Vel13]. The gained flow records show the condition of the network devices, the states of the different protocols from various layers, and consequently the traffic that flows through the network [PC15].

This paper analyses, compares, and evaluates the state of the art and the usability of the IPFIX standard for the industrial automation domain. Since the requirements are increasing in the industry today, the required update rates for the needed information are increasing as well. Therefore, the question arises if an information exchange by IPFIX is possible in the industrial automation domain. The implementation, measurements, and evaluation of this paper can be used to assess the IPFIX standard and whether it can be used for industrial purposes and where the limits are set. This paper is organized as following: Section 2 provides some background information and state of the art, which includes related technologies and an introduction the IPFIX standard and the corresponding architecture. The usage of IPFIX for the industrial domain including the prototypical implementation, the measurements plus evaluation and the security assessment is described in section 3. The last section concludes the work and briefly addresses future work in this area by giving a possible outlook.

2 State of the Art

2.1 Related Technologies

Nowadays the Simple Network Management Protocol (SNMP) is mostly taken as a synonym for network monitoring in general [Bro11]. SNMP was developed in order to collect statistical information about all involved networking devices at one central point in the network. The network devices require so called SNMP network agents which store the desired information inside the device and transfer them to the centralised management station. This can be performed cyclically or manually. The data are collected in management objects which are stored hierarchically in the Management Information Base (MIB). Nevertheless, using SNMP as e.g. counter generates only network traffic statistics and information about the device status. The protocol is not able to support the administrator with the required network traffic information like who has been communicating with whom, when, using which protocol, for how long and how much data has been transferred. Also other technological approaches like The Network Configuration Protocol (NETCONF) in combination with the information modelling language YANG with increased configuration and storage capabilities or the widely used Link Layer Discovery Protocol (LLDP) to collect further device and topology information of the network are not sufficient in order to supply the required statistics about the network traffic. When additional details about the network traffic are required, various other techniques need to be considerated: Deep Packet Inspection (DPI) [WSS16], flow sampling or Packet SAMPling (PSAMP) [Y+13], flow configuration inside Software-Defined Networks (SDN) with e.g. OpenFlow or FlowSense. Nevertheless, the nowadays most typical technological approach for network monitoring is flow export. It can be categorized as a passive monitoring technique and the two most popular solutions are NetFlow and IPFIX.

NetFlow was formerly developed by Cisco as a flexible and light caching technique in order to improve the routing performance of their networking devices and to collect statistical information passively [HCT+14]. Using NetFlow for network monitoring in order to supervise Safety & Security, anomaly detection and QoS control was then introduced later on due to the following advantages: Nearly no capital investment is required because almost all network devices such as routers, switches, servers, or firewalls from various vendors like Cisco, HP, or Microsoft already contain NetFlow capabilities. In addition, to the financial reasons NetFlow measures and reports all Internet Protocol (IP) traffic automatically and scales to different network sizes. Additionally, all different flows need to be identified exactly. Therefore, the packets, which build up a flow are classified with seven characteristics: Source address, source port number, destination address, destination port number, protocol type, Type of Service (ToS), and input logical interface. They are called key fields [HCT+14].

The addresses and port numbers are used to find out about patterns on the network and who is communicating with whom. The protocol type is defined via the used transmission protocol such as the Transport Control Protocol (TCP) or the User Datagram Protocol (UDP). The last characteristic is required for e.g. private IP address spaces where the source address and port number is not sufficient for the flow export. In addition, there are non key fields which can support the flow identification process such as exact timestamps, possible next hops or different subnet masks. By identifying the flows NetFlow is able to create the required flow records,

which are collections of all contained packages. In general, the NetFlow protocol is available in various versions starting with v1 and the current version is v9 providing different functionalities and supporting various use cases or applications. Some versions in between were not even released or are hardly used in real applications. There can be different statements found, which version is the mostly used one, but v5 and v9 are the most important ones [HCT+14][Vel13].

2.2 IPFIX

The IP Flow Information Export Protocol (IPFIX) is an IETF standard for the passive export of flow based network monitoring information for further aggregation and analysis [TB11]. It was first initiated in the RFC 5101 [IET08] and is now developed in the RFC 7011 [IET13]. IPFIX is logically based on the NetFlow v9 protocol and should liberate the vendors from the previously mentioned restrictive constraints of the former NetFlow versions [LSBG13a]. In general it can be described as a bidirectional, transport independent protocol with flexible data representation in order to meet the increased demand in network traffic monitoring and secure data transfer [LSBG13a]. Figure 1 shows the complete arrangement of processes in the IPFIX architecture which is also defined in RFC 5470 [IET09]. This is an example architecture for the usage with IPFIX intended to be a generalization of common measurement infrastructure architectures. The next section will provide additional information about the general architecture of such network monitoring setups [TB11][FY07].

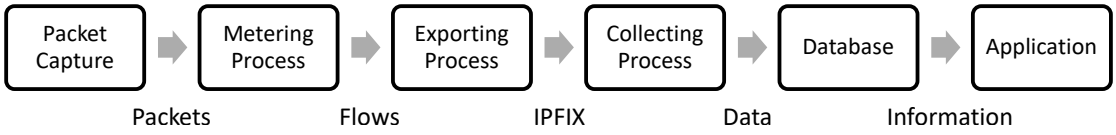

Figure 1: General architecture of IPFIX processes

The composition of IPFIX can be divided into two parts: Structures and dynamics. The first important structure is the **IPFIX message** which represents the basic unit of data transfer and is very similar to the packet format of NetFlow v9. It consists of a 16-byte header and one or more **sets** which have an ID and a variable length and contain the different flow records [TB11]. The header includes a version number, the message length, an export timestamp, the sequence number for missing message detection and an observation domain ID which identifies at which observation point the flow was created. The sets inside the messages can be of one of the three following types [HCT+14]:

1. **Template sets** define the structure and interpretation of the fields in data sets and the layout of the flow records.
2. **Data sets** carry the actual exported data within flow records in the predefined layout by the template sets.
3. **Options Template sets** are used to send additional information to the flow collectors such as monitoring or sampling parameters.

Sets then again contain one or more flow records but the maximum amount of flow records is limited in order to avoid message fragmentation. This is configured during the exporting process before the IPFIX message is send to the data collection point [HCT+14]. Like messages, each set has a 4-byte header containing the set ID and a length field. The set ID refers to another structure named **template**. A template is labelled with a template ID and describes the structure of the specific following data sets by containing a list of multiple Information Elements (IEs). An IE represents a named data field with a specific data type and a specific meaning [TB11]. The IEs are developed and maintained by the Internet Assigned Numbers Authority (IANA) which provides a list of over 450 IEs at the moment in the so called IPFIX Information Element registry [IAN16]. This registry enables vendors and users with interoperability between various software and hardware products on the markets [HCT+14]. IEs consist of a fixed name, numeric ID, description, type, length and a status such as current or deprecated. Nevertheless, it is possible to create enterprise specific IEs by following the basic structure provided by the IANA and adding an enterprise ID at the end. For globally usable new IEs a group of experts from the IANA, the so called IE-Doctors, checks the proposed IEs if they are unique within the IANA IE registry, if they are self contained and if they represent only non proprietary information respectively if they are neither commercially sensitive, experimental or too limited applicable to justify an IANA registration [TB11]. If these

criteria are fulfilled the proposed IE can be added to the registry. Typical are IEs regarding the communication partners, used protocols, important timestamps and additional statistics [TB11][HCT+14].

The second important part of the IPFIX composition contains the dynamics of the protocol by defining the usage of various possible protocols for communication which is required in order to send the constructed IPFIX messages to the data collection. IPFIX supports the usage of three different protocols: The Stream Control Transmission Protocol (SCTP), TCP and UDP. Table 1 shows an evaluation of the possible technological solutions with the help of two ratings ("+- supported / - unsupported) on five different criteria [TB11][T+09][Bro11]:

Table 1: Comparison of transport protocols for IPFIX

Criteria	**SCTP**	**TCP**	**UDP**
Congestion awareness	+	+	-
Deployability	-	+	+
Graceful degradation	+	-	-
Reliability	+	+	-
Secure transport	+	+	-

In conclusion IPFIX is recommended at the moment for flow based network monitoring. It enhances the technical implementation of the system with the greatest possible flexibility regarding flow record layout and export which is desired by the vendors respectively by the users but could not be provided by NetFlow v9. It is even possible to enlarge the flexible structure in this way to integrate syslog or SNMP data such as vendor IDs directly into IPFIX. In addition, there are many optimized IPFIX implementations available on the market and even the newest technologies such as TLS are usable and provide the system with state of the security mechanisms. Therefore, IPFIX must be described more as a generic transport protocol for flexible data structures rather than a pure flow record export tool. This facts make IPFIX usable for network monitoring, QoS and SLA assurance, Safety and Security and even more modern sophisticated applications like smart homes or the Internet of Things.

2.3 IPFIX Flow Monitoring Architecture

2.3.1 Packet Observation

Packet Observation is the first stage of passive network monitoring and therefore the most important stage for a successful implementation. This key stage is described by the process of capturing packets from the network and processing them for the first time for further usage [HCT+14]. It can be again divided into five smaller steps: Packet Capture, Timestamping, Truncation, Packet Sampling, and Packet Filtering.

The initial step is **Packet Capture** which includes reading the packets from the network and collecting them. This is mostly done at the interfaces of packet forwarding devices like a Network Interface Card (NIC) of a router or a switch [HCT+14]. At this step also first checks regarding e.g. checksum or error flags are carried out. After obtaining the packets **Timestamping** occurs. All received packets need to get an accurate timestamp for processing and analysis later on. Most of the data analysis applications require the data in a chronological order for better performance. There are different possibilities for stamping the packets. A very fast solution is the hardware based timestamping with a Field Programmable Gate Array (FPGA) [HCT+14]. But this is only available in special devices which are costly and more difficult to configure. The most used solution is the software based timestamping based on the clock synchronization of the devices with protocols such as the Network Time Protocol (NTP) or the Simple Network Time Protocol (SNTP) [HCT+14]. These technologies provide an accuracy up to 100ms and a simple configuration and implementation. The third step is **Truncation** where the contents of the captured and stamped packets are compared to the configured snapshot length which reduces the obtained amount of data. Traditionally only packet headers are concerned there. With the help of Truncation the resources required for processing the data are reduced consequently resulting in a decrease in computation cycles, used bus bandwidth, internal memory and network traffic later on [HCT+14].The last two steps of Packet Observation are **Packet Sampling** and **Packet Filtering**. Sampling and filtering rules are implemented in order to further reduce the amount of packets which are then forwarded to the Flow Metering and Export stage. Sampling will create a subset of packets which still represents the characteristics of the former set of packets. Some of the lost information such as the total number of bytes can be calculated again at the end. There are different approaches like systematic sampling (deterministic packet selection) or random sampling

(random packet selection). The subsequent filtering removes all unwanted packages from the sample. This can be achieved by preconfigured rules and requirements for the packets which look for specific characteristics inside the packets and remove the ones which do not inherit the wanted property. These properties could be specific IP addresses, protocols or applications. One common way to filter the packet sample is the hash based filtering. The packets hash value is calculated and compared to a predefined range of hash values in order to check if it is wanted or not.

2.3.2 Flow Metering and Export

The second stage of passive network monitoring is the process of flow metering and their export. During this stage single packets are aggregated into flows in the observation points and the collected flow records are exported to the data collection and analysis systems afterwards [HCT+14].

The first step is the **Metering Process** where the flows are collected and aggregated. It consists of various tables named flow caches containing entries which are composed of IEs. The IEs can be either a key or a non-key field. Common key fields are the source or destination address and the port number. Typical non-key fields are flow characteristics such as packet and byte counters. The flow caches represent one active flow for the specific moment in time. New incoming packets are sorted by the properties of the caches IEs and aggregated to the corresponding flow cache. If a match is found and the packet is added to a flow cache its internal packet and byte counters are updated. If no match is found to an existing flow cache the packet is used as the first entry in order to create a new flow cache with its inherited properties [HCT+14]. When designing and implementing new flow caches it is important to select the appropriate one for the specific situation regarding its layout, type and size. In general the flow cache must match the present structure of the IEs in order to support them as good as possible. However with the nowadays high amount of different IEs and their varying usage in network monitoring flow caches need a very flexible and adaptable design. Regarding the type of a flow cache there are three different possibilities at the moment: Regular, immediate and permanent caches [HCT+14]. Regular flow caches are suited for most of today's applications with support for multi packet flows at an acceptable level of performance and advanced cache management tools [HCT+14]. Immediate flow caches should be used for single packet flow applications where only one packet is sampled per flow e.g. in applications with a very high sampling rate. Then the management tools can be neglected and thus the performance can be increased. The third type are permanent flow caches. They can be used in situations where packets are coming in very consistently and continuously. These caches do not expire and export their flow records periodically on a regular basis. The size of the implemented flow caches depends on the available memory of the system, the expected amount of incoming packets, the chosen IEs and the expiration settings [HCT+14].

Flow Sampling and **Flow Filtering** are mostly combined with each other in order to reduce the processing requirements for the exporting step by selecting only subsets of the collected flow records [HCT+14]. Flows are sampled and filtered after they have been aggregated and before they are exported to increase the performance and the throughput of the exporting system and all following stages of passive network monitoring such as data collection and analysis. Similar to the packet observation stage sampling can be done systematically on a deterministic basis or randomly. The filtering of flow records can be achieved by property match filtering (looking at flow record attributes such as specific hosts, subnets or ports) or hash based filtering (calculating the hash value of a specific flow and comparing it to the defined hash value range) [HCT+14].

The **Exporting Process** is the final step of the flow metering and export stage. For this step export and transportation technologies such as the protocol IPFIX are required. The flow records and their stored information are formed into IPFIX messages and exported using the designated flow export communication protocol [HCT+14]. The technological details were already provided in the IPFIX section.

2.3.3 Data Collection

Flow records from various observation and export points in the network must be collected centrally in order to store the data correctly for further analysis. Therefore, the data collection stage is an integral part of the network monitoring setup. The flow records which are collected have already been preprocessed, sampled and filtered by the packet collecting and flow exporting processes in before. Nevertheless, in this stage further data aggregation and compression is performed in order to reduce the required processing power and memory [KSBC10]. The functionality and performance of flow collectors strongly depends on the underlying system and the implemented data storage format because this defines important characteristics such as read and write speed or disk space. In general two storage formats can be distinguished [HCT+14]: Volatile and persistent. Volatile data

storage is based on the memory of the system and is therefore very fast. It is used for caching or preprocessing before writing the data to a persistent data storage which normally has a larger capacity and stores data for a longer period of time but has lower performance regarding reading and writing tasks [HCT+14]. Nevertheless, in nearly all systems both kinds of data storage are implemented and used in order to profit from advantages from both formats. But however the bottleneck between the two storage formats can be described as one present drawback which needs to be regarded in real world applications. Persistent storage can be divided into three different types: Flat files, row-oriented databases and column-oriented databases.

Flat files have a distinctive advantage regarding the required disk space of the system because databases need additional indexes for querying and management of the tables. Also the insertion performance of flat files is very fast due to the fact that new data is just added to the end of the file without advanced management of the data storage. Flat files are mostly stored in single files and are readable by many tools on various platforms. Nevertheless, databases show better performance regarding querying the contained information for specific data due to the implemented indexing which decreases the time and complexity of the search processes. The general performance of the data collection of flow records can be achieved by distribution of data on various storages or adding multiple data collection points to a network using a common management system.

Another important part of the data collection stage is the nowadays commonly expected data anonymization step which implies data obfuscation that ensures anonymity of the traffic users and prevents tracking inside the network. In general flow based monitoring like it is suggested in this work protects privacy better than packet based monitoring because it does not inspect any payloads and consequently the content of the users communication is protected [HCT+14]. Nevertheless, the implementation of the application needs to regard a tradeoff between the anonymization and the information utility. Too strict privacy rules reduce the systems network monitoring functionality and too low privacy rules will result in complaints of the users such as customers.

2.3.4 Data Analysis

Data analysis is the final stage of the passive network flow monitoring architecture. In general there are nearly endless possible applications but monitoring, accounting, traffic profiling, network security, traffic engineering and QoS compliance observance are the main ones which are important for this work [LSBG13a][Z+09]. They can be divided into three main categories: Flow analysis and reporting, threat detection and performance monitoring [HCT+14]. Flow analysis and reporting describes the processes of retrieving comprehensive sets of connection information from the underlying network in order to filter and browse these data, get statistical overviews and receive automatic reports or alerts. Repair misconfigured services, solve routing problems or find high communicating hosts called top-talkers in the network are typical applications inside this category. In general a manual flow data analysis is always possible but not really recommended. Nowadays tools bring a bunch of functionalities with them in order to create automatic reporting by setting specific thresholds for values and defining alert types and messages. The overall goal is to detect problems inside the network as fast as possible and be aware of the present solution approaches in order to tackle the problems as soon as possible [HCT+14]. The second category is threat detection which can be divided into two types of usage. The first type is forensics which describes the analysis of the communication of the network in order to find out about security relevant accidents in the historical data. The second type is the data analysis to detect, prevent and secure the system against various kinds of attacks. Typical attacks are brute force, dictionary or Distributed Denial-of-Service (DDOS) attacks. In addition, systems need to be secured against worm spreading, botnet communication such as the current Mirai botnet and demanding attacks like the Advanced Persistent Threats (APTs) which combine a high computational effort with newest technologies, long term planning and a variety of attack vectors [HCT+14]. Often used characteristics of the flow records for the detection of security related topics are the volume of traffic, suspicious port numbers and the IP addresses of the source and destination hosts. Performance monitoring is the last category of possible data analysis applications. This is used to check the compliance of a Service Level Agreement (SLA) with e.g. customers and if there is an impact on the user experience resulting from a performance drop. Therefore, the QoS parameters of the communication network are monitored and compared with the negotiated and expected values of the signed treaty. Flow based network monitoring is favourable in this application because other approaches such as client based monitoring with software tools or additional hardware requires additional management and configuration effort [HCT+14].

3 IPFIX for the Industrial Automation

3.1 Measurements and Evaluation

Thinking about communication in future industry 4.0 scenarios, communication between machines will be established via plug and play in the future. One important fact in this scenario is the mandatory uniform understanding between the different devices. The currently most common solution for this approach is the use of the Open Platform Communication Unified Architecture (OPC UA) with standardized communication interfaces. While OPC UA uses the client server communication over TCP, another approach is in the standardization process by the OPC Foundation at the moment. The OPC UA PUB/SUB will use the publisher subscriber communication model where also the unconfirmed communication over UDP is possible. Both, TCP and UDP are encapsulated inside IP, so the monitoring of the OPC UA communication can be done by IPFIX.

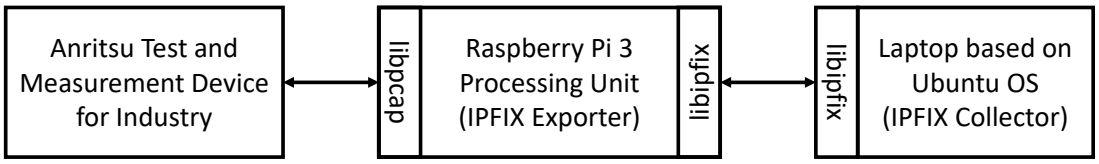

Figure 2: Architecture of the prototypical implementation

The evaluation is based on an analysis of the two libraries libpcap for network traffic capturing and libipfix for passive flow monitoring. With the help of the implemented prototype, which is shown in Figure 2, it is analysed if this combination is suitable for the use inside industrial automation networks. For that reason, the IPFIX exporter will be configured for monitoring of TCP frames. After 100 monitored TCP frames, a flow record will be created and send to the IPFIX collector. Inside Figure 3 the experimental setups are shown and will be explained in the following.

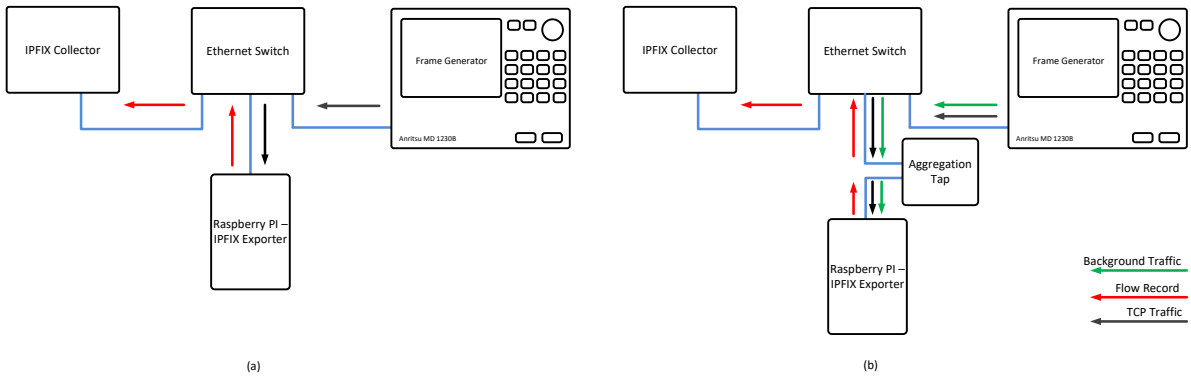

Figure 3: Experimental setup: (a) Measurement 1 (b) Measurement 2

The prototype on the Raspberry Pi containing the IPFIX exporter, the Anritsu network analyzer acting as a frame generator, and the laptop running the IPFIX collector are connected via an Ethernet Switch. With this setup, two different measurements are conducted. The first measurement, shown in Figure 3 part (a) analyses if it is possible to monitor the TCP traffic while running different cycle times, starting with 1 ms up to 64 ms. For that, three different frame sizes (64 byte, 750 byte and 1518 byte) are used. These frames are sent from the Anritsu via the switch to the Raspberry Pi, which captures the frames with libpcap and creates a flow record after 100 monitored TCP frames with libipfix and sends it to the IPFIX collector running on the laptop. Each flow record contains information about the IP address of the sender of the monitored frames, the amount of transmitted octets and packets during the last configured duration (in that case the 100 TCP frames) and the overall amount of monitored packets. To get comparable results by an increasing cycle time, a fixed amount of 100.000 TCP frames will be generated for each test run. That will result in 1000 generated flow records in case of a valid test. The results of measurement 1 are shown inside Table 2.

Table 2: Measurement 1: Different cycle times

Cycle Time	Frame Size	Number of Test Frames	Expected Flow Records	Generated Flow Records
1 ms	64 byte	100.000	1000	752
1 ms	750 byte	100.000	1000	779
1 ms	1518 byte	100.000	1000	808
2/4/8/16/32/64 ms	64 byte	100.000	1000	1000
2/4/8/16/32/64 ms	750 byte	100.000	1000	1000
2/4/8/16/32/64 ms	1514 byte	100.000	1000	1000

The table is split into two sections. In the first half, the results of the measurement with a cycle time of 1 ms, and in the second half, the results for the cycle times from 2 ms up to 64 ms, are shown. During the tests, which are depicted in the second half of the table, all 100.000 frames are monitored in a correct way. Also all the expected 1000 flow records were generated and send with valid information to the IPFIX collector running on the laptop. Looking in more detail to the first half of the table, there is a gap between the expected flow records and the generated flow records. This occurrence can be explained by an analysis of the received frames at the IPFIX collector. From time to time, so called IPFIX partial flow packets are arriving. An IPFIX partial flow packet is characterized by a longer message length of the IPFIX portion of the frame than specified inside the IPFIX message length field. There are also two additional data sets inside such a frame that cannot be decoded by the IPFIX collector. These frames are generated by the IPFIX explorer because it cannot handle the monitored frames, the encoding of the needed information, and the creation of flow records at a cycle time of 1ms. Each time that happens, the IPFIX explorer collects the information from three to four flow records and aggregates them into one. But during these incidents, the information becomes invalid. During the explanation of the measurement setup, the information of the IPFIX flow record are specified by the IP addresses, the amount of bytes and the amount of frames for a generated flow record, and at least the overall amount of packets exchanged between the two devices of the monitored connection. While the overall amount of packets in the last data set is also valid with the arise of partial flow packets, the information inside the dataset two, the amount of packets and the monitored frames inside one flow record are all the time like in a normal IPFIX flow record. Here, some information gets lost. Due to that fact, based on the application that uses the monitored informations, more or less critical missing of information can occur. At the moment and at the current stage of the prototype, valid monitoring with a cycle time of 1 ms or less will not be possible. Monitoring of cycle times from 2 ms to 64 ms is possible and because of that suitable for the operation in industrial automation networks.

Based on the results of measurement 1, the aim of the second measurement, which is shown in Figure 3 part (b), will be the analysis of the situation when there is also some background traffic next to the TCP traffic that is filtered by the IPFIX exporter. Due to the fact that the evaluation should give some findings about using IPFIX inside industrial automation networks, the background traffic will be selected as follows: To simulate the communication that is necessary to fulfil the process that is running inside the production system, some layer two traffic with various lengths and different cycle times per measurement should imitate the process data. Next to that type of traffic, also some network control traffic should be generated by the creation of Link-Layer Discovery Protocol (LLDP) frames and some Discovery and Basic Configuration Protocol (DCP) frames. The LLDP frames will have a frame length of 200 bytes and a send interval of 5 seconds. The DCP frames will also be send with a frame size of 200 bytes and much more as a burst that as cyclic traffic. To check that the IPFIX explorer only takes the generated TCP frames into consideration to create flow record, an aggregation tap should be added on the path between the Ethernet Switch and the Raspberry Pi.

Like mentioned at the beginning of this section, the development of the prototype is still work in progress. Therefore, the described second measurement will be one of the next steps in the future. Before the measurement will be realized, some other requirements, such as the monitoring of more than one connection should be done. After that happens, these and some other measurements with more than one connection, different types of background traffic, and different kinds of data sets inside the flow records are possible. Also some modifications for the monitoring of cycle times of 1 ms should be done within the prototype. Nevertheless, the combination of libpcap and libipfix shows basic potential to use IPFIX for monitoring inside industrial automation networks. Future measurements and a further development will provide additional information on the limitations of IPFIX in the industrial automation and the real benefits of this monitoring approach.

3.2 Security Monitoring using IPFIX

Recent security incidents have revealed, that they are frequently based on using complex, enduring forms of attacks, referred to as Advanced Persistent Threats (APT) [TMSS15]. This allows the attackers to compromise the production network via the Office IT. Protecting the systems by implementing classical preventive countermeasures is less effective, because the attackers are usually very experienced and have an extensive technical know how. Thus they are able to bypass the implemented measures.

In the Office IT, the deployment of Security Information and Event Management (SIEM) systems is a good practice since several years to address this situation. SIEM systems are based on a „detect & resolve“ strategy, instead of completely preventing the attack. Looking a the current trends of the threat landscape for industrial environments, SIEM deployments are also a promising solution in this environment.

A SIEM system consists of a distributed platform, which collects data of different sources. The data is stored and archived in a central data storage. Afterwards the mostly unstructured data is dissected into structured fields by means of textual analysis (normalization), enriched with contextual knowledge (e.g. names of network zones) and provided with an abstract meaning to make it comprehensible for a security expert. These steps are based on the collection of security related information, such as industrial network traffic and device logs, and their correlation. The correlation results allow a fast and efficient response to incidents in real-time [DM15].

However, a major challenge in industrial systems is to get the required information, because logging functionalities of field devices are very limited or not available at all. Furthermore, a huge number of heterogeneous communication protocols must be integrated as well. Both challenges can be addressed by the capabilities of IPFIX [LSBG13b]. The network monitoring data of IPFIX can be used to extend the amount of source data for a SIEM system, which will lead to an improved detection of events. Another relevant use case can be identified in the area of IT forensics. The deployment of IPFIX would allow to collect evidences after an incident whether certain malicious traffic flows were temporary established. This is very important to collect as many insights of an incident as possible to optimize and improve the Security level of the systems. Both use cases are just examples and there are probably several others.

4 Conclusion and Future Work

Nowadays networks in almost every domain are getting bigger, faster, and steadily more complicated. Due to the resulting complex architectures and systems, network monitoring is more and more crucial for the success of the various underlying business processes and general applications. It is mostly used in office environments and huge enterprise communication systems. There it is required for usage based accounting or billing, profiling, network planning, data path reconstruction, or QoS checking for SLA agreements.

In general, there are different approaches for network monitoring. Typical active methods, such as Ping and Traceroute or passive ones like Deep Packet Inspection are too resource intensive, create additional load on the network, and are not privacy sensitive enough by inspecting the direct contents inside the payloads of the captured individual packets. Therefore, passive network monitoring is nowadays mostly performed via flow export which increases the visibility into the traffic across the system greatly and can be used to build up relationships or usage patterns. A flow is a set of IP packets passing an observation point in the network during a certain time interval, such that all packets belonging to a particular flow have a set of common properties. Passive flow monitoring can be examined with various technologies. The most promising ones are NetFlow with its most commonly used versions v5 and v9 or the newest approach IPFIX.

Inside the industrial automation domain, a very similar situation can be found at the moment. The networks there also grow in size and require a great increase in performance. Higher data rates, lower cycle times e.g. in real-time Ethernet based transport protocols are needed. Due to the great amount of network traffic there is a huge potential for additional information export and further data analysis later on. This paper shows the overall usability of the IPFIX protocol for an industrial usage by providing an prototype with a corresponding evaluation. The measurements show the general applicability and the current limitations of the IPFIX protocol inside the industrial automation domain. In addition, the usage of IPFIX for security related solutions and implementations, such as SIEM systems, was discussed and will be persuaded together with the improvement of the prototype in the future work.

References

[Bro11] N. Brownlee. Flow-Based Measurement: IPFIX Development and Deployment. *The Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Communication, August*, 2011.

[DM15] Ralf Schumann Daniel Mahrenholz. Incident response within the SIEM context - A report from the field. In P. Lipp P. Schartner, Hrsg., *DACH Security*, 2015.

[FY07] F. Fatemipour und M. H. Yaghmaee. Design and Implementation of a Monitoring System Based on IPFIX Protocol. In *Telecommunications, 2007. AICT 2007. The Third Advanced International Conference on*, Seiten 22–22, 2007.

[HCT+14] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto und A. Pras. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys Tutorials*, 16(4):2037–2064, 2014.

[IAN16] IANA. IP Flow Information Export (IPFIX) Entities. `http://www.iana.org/assignments/ipfix/ipfix.xml`, 2016. [Online; accessed 19. September 2018].

[IET08] IETF. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. `https://tools.ietf.org/html/rfc5101`, 2008. [Online; accessed 19. September 2018].

[IET09] IETF. Architecture for IP Flow Information Export. `https://tools.ietf.org/html/rfc5470`, 2009. [Online; accessed 19. September 2018].

[IET13] IETF. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. In *lhttps://www.ietf.org/rfc/rfc7011.txt*, 2013. [Online; accessed 19. September 2018].

[KSBC10] T. Kothmayr, C. Schmitt, L. Braun und G. Carle. Gathering Sensor Data in Home Networks. *7th European Conference, EWSN 2010*, Seiten 131–146, 2010.

[LSBG13a] B. Li, J. Springer, G. Bebis und M. H. Gunes. A survey of network flow applications. In *Journal of Network and Computer Applications 36 (2013)*, Seiten 567–581, 2013.

[LSBG13b] Bingdong Li, Jeff Springer, George Bebis und Mehmet Hadi Gunes. A survey of network flow applications. *Journal of Network and Computer Applications*, 36(2):567 – 581, 2013.

[NEWJ17] A. Neumann, M. Ehrlich, L. Wisniewski und J. Jasperneite. Towards monitoring of hybrid industrial networks. *13th IEEE International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, May*, 2017.

[PC15] A. Pekár und M. Chovanec. Survey of the issues surrounding network traffic monitoring. In *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*, Seiten 1–8, 2015.

[T+09] B. Trammel et al. IP Flow Information Export (IPFIX) Applicability - Specification of the IPFIX File Format, 2009. [Online; accessed 19. September 2018].

[TB11] B. Trammell und E. Boschi. An introduction to IP flow information export (IPFIX). *IEEE Communications Magazine*, 49(4):89–95, 2011.

[TMSS15] H. Trsek, D. Mahrenholz, S. Schemmer und R. Schumann. Industrial Security 4.0 - Future challenges and solutions to secure Cyber-Physical Production Systems. In *Markt & Technik Security Symposium 2015 und Industrie 4.0 & Industrial Internet Summit 2015*, October 2015.

[Vel13] P. Velan. Practical experience with IPFIX flow collectors. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013.

[WSJ17] M. Wollschlaeger, T. Sauter und J. Jasperneite. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. In *IEEE Industrial Electronics Magazine Volume: 11 Issue: 1 March*, 2017.

[WSS16] M. Wakchaure, S. Sarwade und I. Siddavatam. Reconnaissance of Industrial Control System by deep packet inspection. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, Seiten 1093–1096, 2016.

[Y+13] C. Yu et al. FlowSense: Monitoring Network Utilization with Zero Measurement Cost. *PAM 2013 14th International Conference*, Seiten 31–41, 2013.

[Z+09] T. Zseby et al. IP Flow Information Export (IPFIX) Applicability. `http://www.ietf.org/rfc/rfc5472.txt`, 2009. [Online; accessed 19. September 2018].