Network Topology Exploration for Industrial Networks

Andreas Paul[✉], Franka Schuster, and Hartmut König

Computer Networks Group, Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany
{andreas.paul,franka.schuster,hartmut.koenig}@b-tu.de

Abstract. Large industrial networks (e.g., plants and grids) are usually characterized by numerous sectors of responsibility and multiple suppliers. Managing these networks is a challenge and requires concrete knowledge of the current network state in terms of device influence and network activities. Here, automated topology exploration is a valuable and very performant measure to provide a wide range of information about devices and their communication relations. Existing exploration methods mostly use active, intrusive methods which have no chance to be applied in sensitive or critical industrial networks. In this paper we present a completely passive approach. It is supplier-independent and provides information that has not been explored before using passive methods.

Keywords: Topology exploration · Industrial networks · Critical infrastructures · Passive network traffic analysis

1 Motivation

In industrial networks a profound knowledge of the current network topology, i.e., the included devices, their roles, and their communication relations with other devices, is essential for the network operator and prerequisite for precise administrative network decisions. Especially, large industrial networks of plants and grids usually involve multiple areas of responsibility and suppliers. Only in rare cases, a well-defined and mandatory documentation process tracks all network configuration changes. Here, topology exploration can provide a direct and reliable picture of the current network configuration and activities as reliable base for various analyses and applications.

Existing methods for network topology exploration either depend on active methods influencing the network by new traffic (which is not acceptable for sensitive or critical industrial networks), are supplier-specific, or lack essential topology information. The proposed approach combines multiple passive methods to provide a high amount of information about devices in the network and their communication independently of the suppliers.

The main contributions of this paper are: (1) we identify multiple (contrary) ways for passively discovering network topologies, (2) we present a fusion of these

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
L.A. Maglaras et al. (Eds.): INISCOM 2016, LNICST 188, pp. 62–76, 2017.
DOI: 10.1007/978-3-319-52569-3_6

methods for gaining a maximum of topology information of a network, (3) we present a graph-based approach to the visualization of the gathered topology information, and (4) we demonstrate the outcome of the approach for different industrial networks.

The remainder of the paper is organized as follows: In Sect. 2, we explain our approach on sole passive topology exploration. The involved methods are applied on a series of experiments, whose results are presented in Sect. 3. After putting our work in the context of other research in Sect. 4, we conclude the paper with an outlook on future research in this field.

2 Methodology

The proposed approach for a sole passive topology exploration is outlined in Fig. 1. It consists of three stages: the *network device discovery*, the *communication discovery*, and the *device information discovery*. The first stage searches for the devices currently included in the observed network domain, whereas the second one tries to enrich the exploration by additional information about the devices. The third stage aims to discover communication relations among the devices. For this, two complementary methods are applied which differentiate in the fact which part of the network messages is used as source of information. The *header inspection* (HI) analyses only the message header, while the *deep packet inspection* (DPI) expands the potential of the header inspection by inspecting the message payload.

2.1 Network Traffic Processing

In this section we describe the function and the implementation of the three basic stages of the network traffic processing unit in detail.

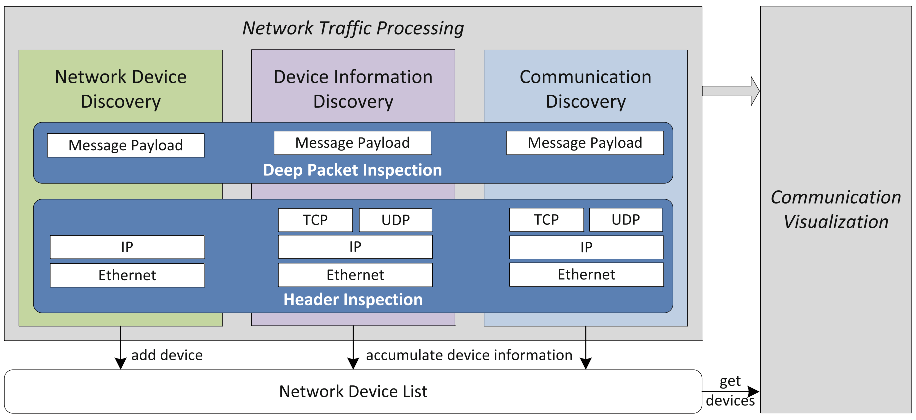

Fig. 1. Topology exploration methodology

Network Device Discovery. As basic method for the network device discovery, we apply header inspection to analyze the source and destination addresses of Ethernet frames and IP packets. However, for indentifying end-to-end communication relations between devices, a strict differentiation between IP packets transported in Ethernet frames and pure Ethernet frames is required. In the case of IP packets, the devices are identified by the IP source and destination addresses. As the Ethernet header stores only the addresses of the point-to-point communication (e.g., routers/gateways or destination device), it is not suitable to use MAC addresses for device identification. Only in case of a pure link-layer communication, it can be assumed that the addresses in the Ethernet header belong to end-to-end communicating devices and can be used for device discovery.

DPI is an advanced method for device discovery because it allows an implicit detection of further network components which are referred to in the message payload. Especially, protocols used for configuration and bootstrap procedures transport information of additional network devices. The principle of implicit device discovery using the ARP protocol is exemplified at the end of this section.

Table 1 summarizes the set of protocols that are currently supported by our device discovery approach. For IP and Ethernet, the header inspection is applied. Implicit device detection is currently performed by decoding ARP, STP, and ICMP. The benefit of the additional application of DPI for device discovery is demonstrated in Sect. 3.2.

Device Information Discovery. Traffic processing also pursues to collect further information about the participating network devices. This can be used to provide operators of industrial networks additional information to recognize unknown devices, e.g., devices identified by (undocumented) MAC addresses. The accumulation of configuration information is also valuable for IT security purposes. It helps to identify existing vulnerabilities and sensitive devices which need a special protection.

Information discovery with header inspection implies a protocol-based assignment of predefined device classes (client, server, router/switch, broadcast/multicast) to the network devices. The Ethernet header is one of the utilized data sources for class assignment (see *(c)* in Table 1). If an Ethernet frame, for example, transports STP (Spanning Tree Protocol) traffic it is assumed that the source device is a switch. This method can be transferred to TCP/IP communication using a port-based protocol detection in combination with a protocol-specific assignment of device classes. Additionally, address-based class assignment is applied. Initially, the vendor of the network device is detected by mapping the first three bytes of the MAC addresses (organizational unique identifier) to the specific vendor (e.g., Cisco) which afterwards is mapped to the device class (e.g., switch/router).

While header-based information derivation is limited to class assignment, the deep packet inspection allows us to derive a more comprehensive set of information. Table 1 indicates that depending on the concrete protocol this also

Table 1. Protocols analyzed in the traffic processing stage

	Protocol	Device discovery		Comm. discovery		Inf. discovery	
		HI	DPI	HI	DPI	HI	DPI
Application layer	NTP						c
	DHCP						c, a
Transport layer	TCP			x		c	
	UDP			x		c	
Internet layer	ICMP		x				c
	IP	x					
Link layer	Profinet IO						a, o
	CDP						a, o
	STP		x			c	o
	LLDP						c, o
	HP						a, o
	ARP		x		x		a
	Ethernet	x		x		c	

[x] Inspection is applied.
[a] Inspection is applied to extract device address information.
[c] Inspection is applied for device class assignment.
[o] Inspection is applied to extract other device configuration data.

may include address information *(a)* and other *(o)* device configuration data (e.g., operating system and symbolic names).

Communication Discovery. A useful method for industrial network analysis is the assessment of communication of the network devices. For this, the set of *communication relations* is collected for each discovered device. We define a communication relation through the triple of source address, the destination address, and the protocol type. An explicit header-based communication discovery comprises in case of IP-based communication a straightforward port-based protocol mapping. In case of non-IP traffic, the communication protocol is derived from the type field of the Ethernet header. Implicit communication detection using deep packet inspection represents a protocol interpretation, which is currently implemented for ARP only (cf. Table 1). The following example explains the idea of implicit communication detection in detail.

Exemplary Message Processing. Since ARP supports all features described in the previous sections, an exemplary ARP request outlined in Fig. 2 is used for clarification. Following the presented methodology of the network traffic processing, this frame is handled as follows:

– **Device discovery:** As an ARP frame is recognized by inspecting the *type* field of the Ethernet header, a new network device *(devA)* is detected based on the

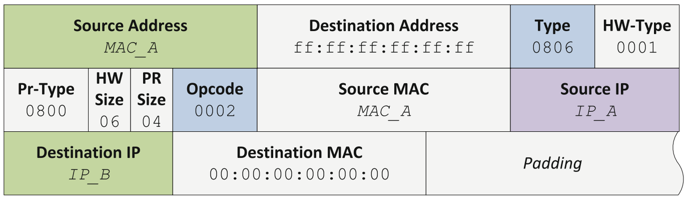

Fig. 2. Exemplary ARP request frame

Ethernet *source address.* The interface function `add_device` (cf. Listing 1.1) either creates a new device or returns its reference, if it is already part of the current network device list. Moreover, the ARP decoder tries to add a further device *(devB)* based on the *destination IP* address, which belongs to the implicitly requested device.

- **Device information discovery:** ARP requests transport the IP address of the requesting device *(source IP)*. Thus, payload inspection enables an IP address assignment to *devA* by use of the function `add_information`.
- **Communication discovery:** Whereas the source address of the message header is used to determine the source of the communication, the implicit requested device *(devB)* rather than the Ethernet destination address (Ethernet broadcast) is interpreted as the destination of the communication relation. This is added during ARP payload processing by use of the interface function `add_relation`. Note that the ARP decoder also determines the specific message type (ARP request) by inspecting the *opcode* field.

```
devA = add_device(device_list, eth.source_address)
process_ARP(devA, device_list, eth.payload)
⟶add_information(devA, arp.source_ip)
⟶devB = add_device(device_list, arp.destination_ip)
⟶add_relation(devA, devB, arp-request)
```

Listing 1.1. Exemplary use of interface functions for network traffic processing

2.2 Visualization of Device Communication

Visualization is considered to be the most effective way to explore the discovered topology. In the following we introduce a graph-based approach to analyze the visualization.

Communication Graphs. We use *communication graphs* for a visual representation of devices and their communication relations. A communication graph is a directed multigraph $G = (V, E)$ with:

- V, a set of vertices, with each vertex $v \in V$ representing a network device discovered in the traffic processing stage.

- E, a set of edges, with each edge $e = ((v_{src}, v_{dst}), M) \in E$ representing a set of communication relations discovered by the network traffic processing between a source device $v_{src} \in V$ and a destination device $v_{dst} \in V$ using the set M of messages consisting of $n \in \mathbb{N}$ different messages.

Visualization Filter. Applying communication graphs results in relative complex visualizations when including the whole set of discovered information. Although this provides a detailed network overview, the analysis should be performed incrementally, e.g., by individual exploration of remarkable devices or specific message types. Hence, *visualization filters* are implemented.

A visualization filter $F = (V_F, M_F)$ consists of a set of vertices V_F, each $v \in V_F$ representing a network device and a set of messages M_F. The application of F to communication graph G results in the subgraph G_{Fw} or G_{Fb} when F works in *whitelist (w)* or *blacklist (b)* mode. While a whitelist filter allows an exclusive visualization of devices and/or messages specified with F, blacklisting operates in the opposite direction for a selective exclusion of specified graph elements. Some characteristics of elements being part of the subgraphs G_{Fw} and G_{Fb} are described in the following.

Characteristics of Network Devices. In case of whitelisting, a discovered device represented by vertex v is specified with V_F, formally described with (1). In contrast, explicit exclusion of a network device from visualization in blacklist mode can be achieved by specifying v with V_F. Thus, v fulfills (2).

$$v \in V \cap V_F \tag{1}$$

$$v \in V \backslash V_F \tag{2}$$

The following characteristics formally describe that v belongs to (is source or destination of) an arbitrary set of communication relations discovered in the network processing stage where at least one communication relation either uses a message specified with M_F (3) or does not use such a message (4) at all.

$$\exists (v_{scr}, v_{dst}, M) \in E : (v_{src} = v \vee v_{dst} = v) \wedge \exists (m \in M : m \in M_F) \tag{3}$$

$$\exists (v_{scr}, v_{dst}, M) \in E : (v_{src} = v \vee v_{dst} = v) \wedge \exists (m \in M : m \notin M_F) \tag{4}$$

With (5), v is characterized to be part of a relation in which the associated communication partner meets (1). To be more restrictive, this assertion can be combined with (3) which results in (6). Whereas these characteristics obviously apply in the context of whitelisting, the adjusted version (7) applies to v in case of communicating to a blacklisted device using a blacklisted message.

$$\begin{aligned} \exists (v_{scr}, v_{dst}, M) \in E : & (v_{dst} = v \wedge v_{src} \in V \cap V_F) \vee \\ & (v_{src} = v \wedge v_{dst} \in V \cap V_F) \end{aligned} \tag{5}$$

$$\begin{aligned} \exists (v_{scr}, v_{dst}, M) \in E : (\ & (v_{src} = v \wedge v_{src} \in V \cap V_F) \vee \\ & (v_{dst} = v \wedge v_{dst} \in V \cap V_F)) \wedge \\ & \exists (m \in M : m \in M_F) \end{aligned} \tag{6}$$

$$\begin{aligned}\exists(v_{scr}, v_{dst}, M) \in E : (\ & (v_{src} = v \wedge v_{src} \in V \backslash V_F) \vee \\ & (v_{dst} = v \wedge v_{dst} \in V \backslash V_F)) \wedge \\ & \exists(m \in M : m \notin M_F)\end{aligned} \tag{7}$$

Characteristics of Communication Relations. Let $e = (v_{src}, v_{dst}, M)$ be a set of communication relations which belongs to the set of visualized edges when applying F to G. Set e is a subset of a set of discovered relations $e^{'} = (v^{'}_{src}, v^{'}_{dst}, M^{'})$ with $v_{src} = v^{'}_{src}, v_{dst} = v^{'}_{dst}$ and $M \subset M^{'}$ with each $m \in M$ matching the characteristics specified with the following assertions.

The first assertions characterize for each e that its communicating devices belong to the set of visualized devices when F operates in whitelist (8) or blacklist (9) mode. As no requirements on the messages are made, all elements of $M^{'}$ are included in M.

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : & (v_{src} = v^{'}_{src} \wedge v^{'}_{src} \in V_{Fw}) \wedge \\ & (v_{dst} = v^{'}_{dst} \wedge v^{'}_{dst} \in V_{Fw}) \wedge \\ & M = \{m | m \in M^{'}\}\end{aligned} \tag{8}$$

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : & (v_{src} = v^{'}_{src} \wedge v^{'}_{src} \in V_{Fb}) \wedge \\ & (v_{dst} = v^{'}_{dst} \wedge v^{'}_{dst} \in V_{Fb}) \wedge \\ & M = \{m | m \in M^{'}\}\end{aligned} \tag{9}$$

With (10) and (11), e is characterized by the exclusively use of messages specified with M_F or rather by the use of not blacklisted messages. The two most restrictive assertions on a set of communication relations simply combine (8) and (10) to (12), respectively (9) and (11) to (13).

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : & (v_{src} = v^{'}_{src} \wedge v_{dst} = v^{'}_{dst}) \wedge \\ & M = \{m | m \in M^{'} \cap M_F\}\end{aligned} \tag{10}$$

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : & (v_{src} = v^{'}_{src} \wedge v_{dst} = v^{'}_{dst}) \wedge \\ & M = \{m | m \in M^{'} \backslash M_F\}\end{aligned} \tag{11}$$

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : (\ & (v_{src} = v^{'}_{src} \wedge v^{'}_{src} \in V_{Fw}) \wedge \\ & (v_{dst} = v^{'}_{dst} \wedge v^{'}_{dst} \in V_{Fw})) \wedge \\ & M = \{m | m \in M^{'} \cap M_F\}\end{aligned} \tag{12}$$

$$\begin{aligned}\exists(v^{'}_{src}, v^{'}_{dst}, M^{'}) \in E : (\ & (v_{src} = v^{'}_{src} \wedge v^{'}_{src} \in V_{Fb}) \wedge \\ & (v_{dst} = v^{'}_{dst} \wedge v^{'}_{dst} \in V_{Fb})) \wedge \\ & M = \{m | m \in M^{'} \backslash M_F\}\end{aligned} \tag{13}$$

Characteristics of $\mathbf{G_{Fw}}$ and $\mathbf{G_{Fb}}$. For describing the characteristics of the resulting subgraphs when applying a visualization filter F to a graph G, four cases have to be considered with regard to the composition of F:

1. $V_F = \emptyset \wedge M_F = \emptyset$: In case of an empty filter with no elements specified with V_F and M_F the subgraphs directly correspond to G.

Table 2. Subgraph characteristics

Case	G_{Fw} (Whitelist mode)	G_{Fb} (Blacklist mode)
1	$V_{Fw} = \{v \mid v \in V\}$	$V_{Fb} = \{v \mid v \in V\}$
	$E_{Fw} = \{e = (v_{src}, v_{dst}, M) \mid e \in E\}$	$E_{Fb} = \{e = (v_{src}, v_{dst}, M) \mid e \in E\}$
2	$V_{Fw} = \{v \mid (1) \vee (5)\}$	$V_{Fb} = \{v \mid (2)\}$
	$E_{Fw} = \{e = (v_{src}, v_{dst}, M) \mid (8)\}$	$E_{Fb} = \{e = (v_{src}, v_{dst}, M) \mid (9)\}$
3	$V_{Fw} = \{v \mid v \in V \wedge (3)\}$	$V_{Fb} = \{v \mid v \in V \wedge (4)\}$
	$E_{Fw} = \{e = (v_{src}, v_{dst}, M) \mid (10)\}$	$E_{Fb} = \{e = (v_{src}, v_{dst}, M) \mid (11)\}$
4	$V_{Fw} = \{v \mid (1) \wedge (6)\}$	$V_{Fb} = \{v \mid (2) \wedge (7)\}$
	$E_{Fw} = \{e = (v_{src}, v_{dst}, M) \mid (12)\}$	$E_{Fw} = \{e = (v_{src}, v_{dst}, M) \mid (13)\}$

2. $V_F \neq \emptyset \wedge M_F = \emptyset$: A sole device based filter includes (whitelist) all devices specified with V_F and the belonging communication relations into the visualization, respectively excludes them in case of blacklisting.
3. $V_F = \emptyset \wedge M_F \neq \emptyset$: A message based filter operating in whitelist (or blacklist) mode exclusively considers (or excludes) communication relations that are based on messages specified with M_F and the associated communication partners.
4. $V_F \neq \emptyset \wedge M_F \neq \emptyset$: Device and message based filtering can be combined which results in the most restrictive set of visualized graph elements.

Table 2 summarizes the exact composition of $G_{Fw} = (V_{Fw}, E_{Fw})$ and $G_{Fb} = (V_{Fb}, E_{Fb})$ for each of these cases by implicit specifications of the sets of vertices and edges. For this, the characteristics of their elements are indicated by use of (1)–(13).

Exemplary Communication Visualization. The methodology of device communication visualization is exemplarily explained by the graph presented in Fig. 3. The graph consists of four vertices, three of them represent real network devices labeled with the discovered MAC[1] and/or IP addresses. The virtual device indicated by the dashed frame line results from using a non-interpreted broad or multicast address (e.g., the spanning tree multicast address in case of STP traffic) as destination of a communication relation. Figure 4 shows the graph that results when applying a whitelist filter with the following characteristics:

– $V_F = (switch/router)$: The exclusive visualization of the switch is achieved by using the corresponding device class as a device-based filter. Alternatively, devices can also explicitly be specified with V_F by using MAC or IP addresses. Devices matching a device-based filter are indicated in the graph by green-filled vertices.

[1] OUI lookup is used for integration of the manufacturer or the name of well-known broad and multicast addresses into the MAC address label.

– $M_F = (arp)$: A message filter is used for an exclusive visualization of ARP frames. Further possibilities to specify messages of the subgraph are the indication of specific protocol message types (e.g. ARP request) or to filter several protocol classes, e.g., TCP or UDP to indicate the transport protocol.

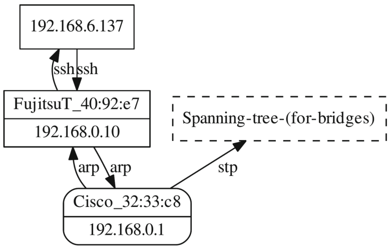

Fig. 3. Complete graph

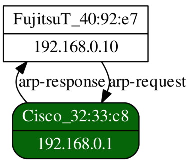

Fig. 4. Subgraph

3 Application and Evaluation

In order to evaluate and test our passive topology exploration approach we applied three datasets (cf. Table 3) from industrial networks. In our evaluation we especially focus on the benefit of applying deep packet inspection and communication visualization.

3.1 Applied Datasets

The applied datasets differ in size and deployment purpose. Dataset 1 is derived from a network used to operate a double-unit lignite power plant. The traffic sensor was placed on one of eight main switches interconnecting several process control and expert systems. Beside standardized IT protocols the communication runs over proprietary vendor specific protocols on top of TCP/IP. Full access to the traffic of the observed domain was realized by port mirroring.

The second dataset was captured from a flexible lignite conveyer system situated in an opencast mine used for bridging the way between the excavator and long-distance vehicles transporting the lignite out of the mine. The sensor was connected to a mirror port of a switch its interconnected devices basically communicate via an Industrial Ethernet based field bus, namely Profinet IO.

Source of dataset 3 is a test system of a power distribution grid. The observed network traffic contains telecontrol communication between a control center, a transformer station, and a local grid using the IEC 60870-5-104 protocol on top of Ethernet as well as IP. Again, the sensor had access to the complete traffic passing the main network switch.

Table 3. Characteristics of the datasets

	Dataset 1		Dataset 2		Dataset 3	
	Total	bc/mc	Total	bc/mc	Total	bc/mc
Size [MB]	1,007	1.2	1,126	1.2	112	84.3
# packets	3,637,652	12,064	5,775,465	7,279	716,655	550,348
Duration [min]	25		19		941	
# devices	133		33		12	

Of course, access to a mirror port cannot be guaranteed in each scenario, especially when applying the exploration to networks which operate in critical infrastructures. For simulating a traffic sensor connected to a default switch port, we extracted broad- and multicast messages from the total captures. The characteristics of the resulting datasets (bc/mc) are also summarized in Table 3.

3.2 Network Device Discovery

For evaluating the device discovery, we first investigate the capabilities of implicit device detection, followed by indicating the potential of a comprehensive topology exploration regarding the proportion of network components discovered by use of passive traffic analysis. The results discussed are presented in Table 4. Here, all disclosed numbers of devices exclusively refer to discovered devices with an assigned IPv4 unicast address.

DPI Benefit. In order to demonstrate the benefit of implicit device detection using deep packet inspection the number of devices detected only by header inspection (n_{HI}) is compared with number of devices detected additionally by means of DPI (n_{DPI}). For this, Table 4 indicates the *DPI benefit* which is defined as follows:

$$DPI\,benefit = \frac{(n_{DPI} - n_{HI}) \cdot 100\%}{n_{HI}} \tag{14}$$

The relatively low DPI benefit in comparison to the total captures with a maximum of about 27% (three more detected network devices in dataset 3) can be explained by the position of the traffic sensors, which were connected to a mirror port of a main network router or switch, respectively. An exclusive consideration of broad- and multicast traffic shows a substantial increase of the DPI benefit which raises up to 1100% in case of dataset 2. Moreover, it should be noted that each device discovered by the full capture could also be identified in the broad- and multicast traffic of datasets 2 and 3.

Device Detection Rate. The underlying infrastructure of the datasets used in the evaluation is characterized by various interconnected networks. To determine meaningful results concerning the device detection rate the sets of *detectable*

Table 4. Discovered network devices

	Dataset 1		Dataset 2		Dataset 3	
	All	bc/mc	All	bc/mc	All	bc/mc
n_{HI}	96	47	21	2	11	3
n_{DPI}	101	62	24	24	14	14
DPI benefit [%]	5.21	31.91	14.29	1100.00	27.27	366.67
n_d	88		22		11	
DDR [%]	81.82	61.36	90.91	90.91	90.91	90.91

devices were determined first. These are subsets of documented devices that belong to the same network as the device on which the traffic sensor was placed. The percentage ratio between detected devices, which are part of the set of detectable devices, and the total number of detectable devices (n_d) is indicated as *device detection rate (DDR)*. As Table 4 indicates, passive device discovery reaches a detection rate up to 90% for the datesets 2 and 3, even if only broad- and multicast messages are considered.

Note that due to differences in quality and completeness of the documentation required to determine the sets of detectable devices, DDR is not seen as a clear criteria for the evaluation. Instead, the occurrence of undocumented devices can be indicated by comparing the sets of detected and detectable devices. As the number of discovered devices n_{DPI} exceeds the number of detectable devices, an out-dated documentation has to be assumed for each of the three datasets.

3.3 Network Analysis Using Communication Visualization

Now we discuss exemplarily the results from applying network topology visualization approach using communication graphs (cf. Sect. 2.2) to the datasets 1 and 2. Due to demands of the network operators, all addresses mentioned in this section had to be anonymized as follows: The pattern $MAC\_ < number >$ ($IP\_ < number >$) is used to represent a unique MAC (IP) address, whereby equal addresses within a network are represented by equal numbers.

Analysis of Individual Device Communication. The first scenario presents a practical procedure for analyzing devices to which special attention should be paid, e.g., unknown (undocumented) devices discovered by automated topology exploration. It is obvious that in a first step the address of the device to be investigated should be used to whitelist its whole communication. Figure 5 shows the resulting communication graph by use of an exemplary undocumented IP address *IP_04* discovered from dataset 1.

The exclusive discovery of ARP communication indicates, that the assignment of the IP address to the network interface with *MAC_04* was provided by DPI. As presented by the node label, multiple IP addresses are assigned to the

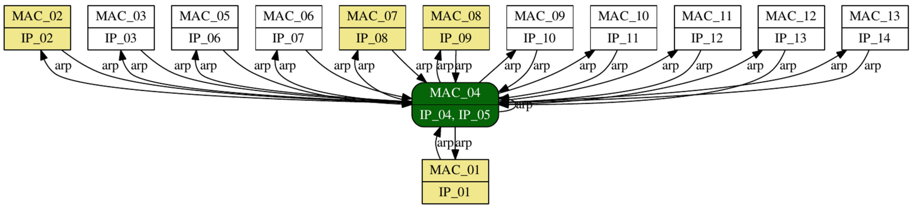

Fig. 5. Communication graph resulting from *device* whitelisting

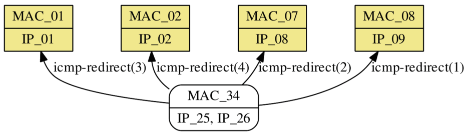

Fig. 6. Communication graph resulting from *protocol* whitelisting

device. The additional IP address *IP_05*, which is also not part of the documentation, was also announced by the device using ARP request frames[2].

The considered device is classified as "switch/router", even though exclusive ARP relations are discovered. In this case, role assignment was applied by the ICMP decoder, more precisely by interpreting the address of a preferable gateway announced from a router using ICMP redirect packets. Applying an edge (protocol) based whitelist filter for an exclusive consideration of ICMP redirect messages results in the graph presented in Fig. 6. The (documented) router with the IP addresses *IP_25* and *IP_26* sends redirect messages to four devices. All of them are also addressed by the undocumented router, which can immediately be recognized by use of the yellow-filled node marking.

Thus, beside identifying an undocumented router the analysis of an individual device also reveals a security lack inside the observed network. As ICMP redirect messages can easily be exploited to launch man-in-the-middle attacks, it is recommended not to use (send and process) them any more.

Analysis of the Complete Network Communication. This scenario evaluates the capability to distinguish between the different roles of the devices by analyzing their communication patterns. For this, the whole unicast communication discovered in the network represented by dataset 2 is represented by the graph shown in Fig. 7. The graph already indicates significant differences in the communication patterns regarding the number of addressed communication partners. Most communication partners by far are addressed by device *MAC_01*. The 17 addressed components include 16 decentralized peripheral devices and one PC

[2] Because of space limitations the edges of the graph are not labeled with the full protocol message names. All outgoing arcs of the device *IP_04* represent ARP request frames, while incoming arcs belong to the respective responses.

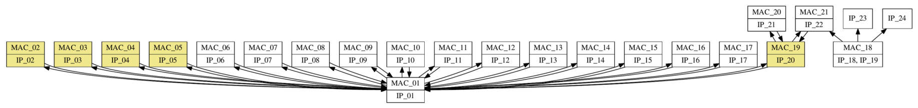

Fig. 7. Communication graph summarizing the unicast communication of a network.

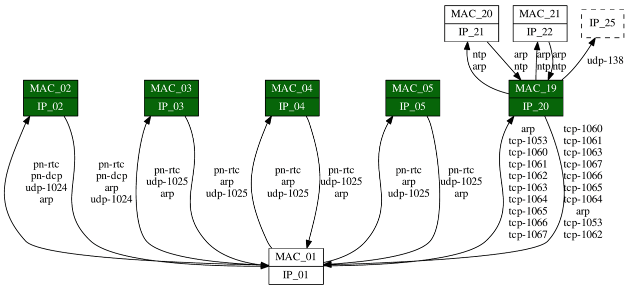

Fig. 8. Communication graph presenting relations of different roles of devices.

used for diagnostics. Decentralized peripheral (devices *MAC_02-17*) are solely identified by the data exchange with its superordinated PLC (device *MAC_01*).

These assumptions regarding the device roles can be confirmed by a more detailed view on the communication patterns by taking the communication protocols into account. For this, Fig. 8 depicts the exclusive communication from devices *MAC_02-05*[3] and the PC *(MAC_19)*. The field layer communication is essentially characterized by a cyclic real-time transfer of process data using the Profinet IO protocol *(pn-rtc)* and a request/response-based query of specific PLC slots via Profinet IO Context Management running over UDP (ports 1024 and 1025). Optionally, devices exchange configuration information by use of the Profinet Discovery and Configuration Protocol *(pn-dcp)*. In contrast, several applications running over TCP are addressed through the communication from the diagnostic PC to the PLC.

4 Related Work

Related work of automated topology exploration comprises a broad research area, but the number of comparable approches is quite limited due to the pure passive character of our approach. Most of the related approaches rely on active techniques, e.g., for device discovery via ICMP [12,13], MAC probe sending [7], and querying intermediate systems with SNMP (e.g., [3,5,8,12]). Partially, the topology exploration even requires specific software installation on the participating network hosts [2,7]. The main advantage of active methods is the possibility to determine wired connections between network devices that enables a

[3] Due to space limitations, these devices are chosen to represent the communication patterns of the decentralized periphery. The remaining devices *MAC_05-MAC_17*, however, exhibit similar relations to device *MAC_20*.

reconstruction of the real physical topology. Although this is a valuable skill, the generation of additional network traffic and the installation of further software is not acceptable in the considered area of industrial networks.

Approaches on passive network discovery are very rare. In [4] an algorithm is proposed that uses hop-counts for distance measurements. As the associated information (Time-to-Live value) is modified by network routers, this method can be used exclusively for topology discovery on the internet protocol layer, but it does not work solely on the data link level. Closest to our approach is the network analyzer in [9] that also supports passive device and communication discovery. Header inspection in the internet protocol layer is applied in the two cases. This is not applicable to handle pure Ethernet frames. Since real-time variants of the Industrial Ethernet and other vendor-specific protocols are implemented directly on the data link layer, the processing of the Ethernet protocols is indispensable for a detailed analysis of local industrial networks. Furthermore, we differentiate ourselves from existing solutions by the use of protocol decoders. To the best of our knowledge, our approach exclusively applies deep packet inspection to maximize the potential of passive topology exploration.

5 Final Remarks

Automated network topology exploration is a challenging issue, especially when avoiding interferences resulting from active techniques has top priority. In this paper, we have presented a novel method for passive traffic processing that makes topology discovery suitable for application in the industrial area. Our future research will focus on utilizing the collected topology information to improve industrial network security by means of network-based intrusion detection. In a first step, the set of communication relations can be transformed into a rule set processed by a signature-based intrusion detection system (e.g., Snort [1]). Due to the homogeneity of industrial traffic [6], this provides an initial protection by detecting changes in the topology caused by unknown devices or communication relations. We further plan to combine this rule-based analysis with a more advanced approach on anomaly detection based on machine learning methods presented in [10]. To this end, we will focus on applying rule-based IDS alerts as a replacement of a feature subset with the aim to reduce the feature space and thus increasing the current detection accuracy [11].

Acknowledgements. The authors gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) via the projects INDI (funding code: 16KIS0156) and SICIA (16KIS0158K).

References

1. Snort: Network intrusion detection system (2016). https://www.snort.org
2. Black, R., Donnelly, A., Fournet, C.: Ethernet topology discovery without network assistance. In: Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP 2004), Berlin, Germany, 5–8 October 2004, pp. 328–339 (2004)

3. Breitbart, Y., Garofalakis, M.N., Jai, B., Martin, C., Rastogi, R., Silberschatz, A.: Topology discovery in heterogeneous IP networks: the NetInventory system. IEEE/ACM Trans. Netw. **12**(3), 401–414 (2004)
4. Eriksson, B., Barford, P., Nowak, R.D., Crovella, M.: Learning network structure from passive measurements. In: Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, IMC 2007, San Diego, California, USA, 24–26 October 2007, pp. 209–214 (2007)
5. Gobjuka, H., Breitbart, Y.: Ethernet topology discovery for networks with incomplete information. IEEE/ACM Trans. Netw. **18**(4), 1220–1233 (2010)
6. Hadžiosmanović, D., Simionato, L., Bolzoni, D., Zambon, E., Etalle, S.: N-Gram against the machine: on the feasibility of the N-Gram network analysis for binary protocols. In: Balzarotti, D., Stolfo, S.J., Cova, M. (eds.) RAID 2012. LNCS, vol. 7462, pp. 354–373. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33338-5_18
7. Kienzle, D.M., Evans, N.S., Elder, M.C.: NICE: endpoint-based topology discovery. In: Cyber and Information Security Research Conference, CISR 2014, Oak Ridge, TN, USA, 8–10 April 2014, pp. 97–100 (2014)
8. Lowekamp, B., O'Hallaron, D.R., Gross, T.R.: Topology discovery for large ethernet networks. In: SIGCOMM, pp. 237–248 (2001)
9. Moussadek-Kabdania, A., Soilli, A.: Grassmarlin, an open-source tool for passive ICS network mapping (2016). http://www.securityinsider-solucom.fr/2016/03/en-grassmarlin-open-source-tool-for.html
10. Schuster, F., Paul, A.: A distributed intrusion detection system for industrial automation networks. In: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation, ETFA 2012, Krakow, Poland, 17–21 September 2012, pp. 1–4. IEEE (2012)
11. Schuster, F., Paul, A., Rietz, R., König, H.: Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. In: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, 7–10 December 2015, pp. 83–90. IEEE (2015)
12. Wang, Y., Li, D., Han, C., Zhu, Z.: Research and application on automatic network topology discovery in ITSM system. In: Proceedings of the 9th International Conference on Hybrid Intelligent Systems, Shenyang, China, 12–14 August 2009, pp. 336–340 (2009)
13. Yao, B., Viswanathan, R., Chang, F., Waddington, D.G.: Topology inference in the presence of anonymous routers. In: IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA, March 30 - April 3 2003 (2003)