OPC UA extension for IP Auto-Configuration in Cyber-Physical Systems

Markus Rentschler
Business Unit Networking
Balluff GmbH
Neuhausen, Germany
markus.rentschler@balluff.de

Henning Trsek
Industrial Security
rt-solutions.de GmbH
Cologne, Germany
trsek@rt-solutions.de

Lars Dürkop
Institute Industrial IT (inIT)
Ostwestfalen-Lippe UAS
Lemgo, Germany
lars.duerkop@hs-owl.de

***Abstract*—The ability to remotely discover and configure devices in an automation network without the need for prior knowledge of the network topology or the identities of hosts and servers is a key requirement for industrial networks and cyber-physical systems. The Dynamic Host Configuration Protocol (DHCP) as a UDP-based standardized network protocol became very successful in computer networking and also to some extent in industrial networking, but it conceptually allows only device-initiated parameter assignment. This has led to the development of several other device configuration protocols that allow host-initiated device discovery and configuration. In this work, a survey on existing protocols is performed. Based on the results, a future solution is proposed as an extension of the OPC UA protocol suite. The proposal is able to meet the industrial needs for both device- and host-initiated operation with minimal invasive implementation.**

I. Introduction

Recently, a paradigm shift from mass production to mass customization could be observed within the manufacturing industry as a response to rapidly changing customer demands. Since changing the manufacturing process requires a lot of manual configuration steps and a high amount of expensive human resources, reconfigurable manufacturing systems have been introduced and industrial automation systems must also provide suitable solutions to support this new paradigm and its flexibility. For example, modules and machines can be added or removed from the production process to achieve new functionalities enabling a response to unforeseen requests.

The SmartFactory OWL (SFOWL)[1] is an example for a reconfigurable manufacturing system based on a modular mechatronic approach and considered as the application scenario for this work. The current architecture consists of flexible modules, which are intelligent, autonomous production units and can be referred to as Cyber-Physical Systems (CPS) following the concepts of Industry 4.0. The modules feature self-configuration and self-diagnosis capabilities to be able to adapt to a changing environment autonomously. Hence, they can be added and removed in order to adapt the overall manufacturing process. Each module is equipped with a local programmable logic controller (PLC) and IO devices to acquire its local sensor values and to set its actuators. The modification can be distinguished in the use-cases (i) initial setup of the system, (ii) reconfiguration by adding a completely new module, (iii) reconfiguration by replacing a module, and (iv) maintenance by exchanging modules or devices. Whenever the SFOWL is modified, the modules must be integrated into the whole system. As the first step in the auto-configuration process of an industrial module or device, after it is newly connected to the network, an IP address must be assigned to ensure its IP connectivity, and establish a communication channel to the remaining plant. Depending on the identified use case, this basically requires:

- Remote discovery and configuration without prior knowledge of the network topology and device identities
- Deterministic IP address assignment, i.e. allocated addresses must be unique and assigned in a given time to avoid interruptions of the manufacturing process
- Assignment either initiated by a central engineering host, or by a device or module
- Global assignment of addresses without being limited to a single network segment or subnet

Up to now, well-known standard protocols like DHCP or other specific means, such as a dedicated service incorporated in a Real-time Ethernet protocol suite are used for this. Besides this, a range of standard and proprietary protocols have been implemented by different vendors to fulfill this task. Automation engineers with heterogeneous product installations are in consequence required to manage a diverse tool-chain to operate these protocols, which does not contribute to an easy usability in such environments. Although initiatives like Industry 4.0 claim as main feature the integrated capability of auto-configuration [1], they do not yet define themselves an integrated solution for initial IP parameter assignment. Thus, the automation engineers still have to maintain the mentioned diverse tool-chains for the simple task of initial IP address assignment, which is an absolute necessity when commissioning devices in an automation plant.

[1] http://www.smartfactory-owl.de

 DOI: 10.1109/INDIN.2016.7819128

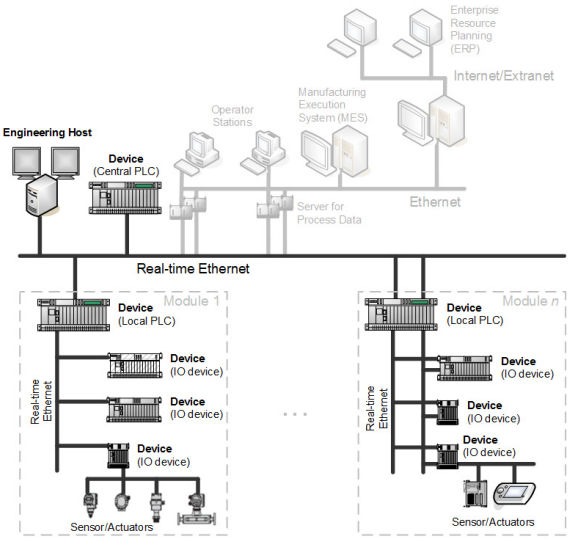

Fig. 1. SFOWL architecture, topology and terminology

To overcome the aforementioned problem, in this work an improvement path to the current situation is defined by proposing a solution approach towards a standard initial IP assignment method for Industry 4.0 and CPS, which is integrated into the OPC UA discovery mechanism. Since such parameter assignment protocols are usually based on client-server architectures, the placement of each logical entity depends on the direction of the protocol operation. Since communication direction may change, a unified architectural definition with an associated terminology used throughout this paper is defined in the context of an industrial network in Fig. 1, where a **Device** defines the entity that requests or is assigned a set of IP parameters and an **Engineering Host** defines the entity that grants the assignment of a set of IP parameters. Both Device and Engineering Host can be any kind of TCP/IP-capable device in a network and even incorporate both functions in certain hierarchical directions of the network.

The remainder of this paper is organized as follows. In chapter II, an overview on the development of auto-configuration mechanisms as well as related work is presented. The current adoptions of DHCP in industrial networks and other approaches are discussed in chapter III. In chapter IV, a possible solution based on an extended DHCP is presented, whereas in chapter V, the extension of OPC UA is presented. Finally, chapter VI concludes this contribution.

II. Overview and Related Work

A first approach towards IP address auto-configuration within the TCP/IP protocol suite was the "Reverse Address Resolution Protocol" (RARP) as described in [2]. Since RARP operates on the data link layer, it could only be implemented with access to this lower communication layer, making implementations hardware dependent and required a dedicated server for each network segment.

For these reasons, RARP has been rendered obsolete by the Bootstrap Protocol (BOOTP), as defined in RFC 951, BOOTP supported a much greater feature set than RARP and instead of working on the link layer level, was UDP/IP-based, working on UDP ports 67 and 68. Making use of UDP broadcasting, the BOOTP protocol cannot pass through routers, but the *relay agent* concept also introduced in RFC 951 supports the forwarding of BOOTP packets across networks, allowing one central BOOTP server for different IP subnets. Although BOOTP was able to configure other parameters than the IP address, it still needed manual intervention on the server host to add configuration information for each individual client device and could not provide a mechanism for reclaiming disused IP addresses.

To overcome the limitations of BOOTP, DHCP was first defined in RFC 1531. After some modifications RFC 2131 was released and defines the standard for IPv4 networks until today. DHCP uses the same message format as BOOTP, and BOOTP client devices can be served by DHCP servers. To support the IPv6 protocol, DHCPv6 was introduced and documented in RFC 3315. Both DHCPv4 and DHCPv6 are extensible protocols with many additional options already defined in several RFC's.

A link-local address range for IPv4, 169.254.0.0/16, is defined in RFC 3330. Accordingly in IPv6, every interface automatically assigns a local-link address in the block fe80::/10. Since no auto-configuration for these addresses were available, the Automatic Private IP Addressing (AutoIP) was specified. The IETF Zeroconf working group defined in RFC 3927 a standard functionality, called "*Dynamic Configuration of IPv4 Link-Local Addresses"* [2]. Due to its non-deterministic automatic approach and some other flaws as analyzed by Gebremichael et al. [3], the local-link addressing could not gain acceptance for industrial networks.

Guttman addressed an approach of a zero configuration protocol for IP networks in [4]. Zero configuration protocol will likely simplify network configuration by reducing manual settings of IP networks with modifying and operating IP hosts for local communication, but only within a single network segment.

Swales introduces in [5] an Auto-IP software tool which supports auto configuration for industrial network by using a combined BOOTP/DHCP and SNMP approach, called "Network Vision Auto-IP assignment".

In [6] Norman discusses the usage of DHCP to intelligently assign IP addresses in industrial network applications with different network topologies and scenarios.

Weniger et al. [7] presented PACMAN as a new approach for distributed IP address auto-configuration which follows a hybrid approach to provide an efficient address assignment for mobile ad-hoc networks. A novel auto-configuration approach for mobile ad-hoc networks is also addressed in [8]. The proposed protocol has a tree structure that combines the advantages of distributed and centralized approaches to assigns an IP address to the un-configured nodes in the network in a reliable process with low communication overhead. The protocol depends mainly on the unicast messages and does not rely on message flooding that consumes a lot of channel bandwidth. Similarly, Bernardos et al. collected in [9] different auto-configuration mechanisms and assessed the results with classifications accordingly.

Most of these works aimed to address IP auto-configuration mechanisms based on DHCP which is not able to completely cover the auto-configuration requirements for complex industrial networks. Nevertheless, our requirements

are partially answered by these works, but still there is no solution addressed for some of the requirements. Specially, for central address re-assignment, global address assignment and authentication, there are no solutions identified so far.

III. IP Address Assignment in Industrial Networks

The IP address assignment in industrial networks can be either manual or automatic. In the first case, the address is assigned manually by a human operator using a dedicated tool, as discussed in Sec. III.B. The automatic assignment is done whenever the automation device or module is connected without requiring human intervention. Obviously, the latter is more relevant in this work. The automatic IP address assignment can be further distinguished in two different approaches, (i) the assignment is driven by the new node, and (ii) the assignment is initiated by a central entity. Both approaches are discussed in the remainder of this section. They are both relevant and the selection usually depends on the specific use case and its requirements.

A. Dynamic Host Configuration Protocol

The BOOTP and DHCP as semi-automatic mechanisms in terms of configurability of assignment policies were only to some extent accepted in industrial networks

DHCP was designed as a "Plug 'n Play" (PnP) like approach, enabling client devices to acquire a network address in the initial phase of joining a network without personal intervention of a network administrator. However, for industrial networks several insufficiencies were identified. Using dynamic IP address ranges, there is no determinism in how IP addresses are assigned to specific devices in the network topology. This is especially problematic in the case of faulty device replacement where a broken device is replaced with an identical new device. In this case the assignment of precisely the same parameter set through DHCP is required, but the MAC address as the only reliable device identifier has changed. Several approaches were defined to solve this problem, a taxonomy is presented in [10]. Popular examples are the usage of DHCP option 82 and distributed DHCP servers that operate on a per port basis. These approaches make in fact use of the topological position of a device in a network, which does not change when just a single device is exchanged.

Since DHCP is based on device-initiated communication, an inherent problem is the lack of possible active operations from engineering host to device [11], which is often desired for manually controlled parameter assignments in the initial commissioning phases of automation networks.

DHCP is defined as optional for PROFINET, but not even implemented in many PROFINET stacks. For the ODVA's Ethernet/IP protocol suite, DHCP is defined as mandatory and the use of Option 82 recommended. For other TCP/IP-based automation networks, i.e. based on Modbus/TCP, BOOTP or DHCP form usually an integral part.

B. Discovery and Configuration Protocols

The inherent usability flaws of DHCP in the initial device deployment phase have led to the development of other discovery and configuration protocols for industrial devices. This section will deal with the characteristics of these protocols, that specifically incorporate network scanning and static address assignment capabilities.

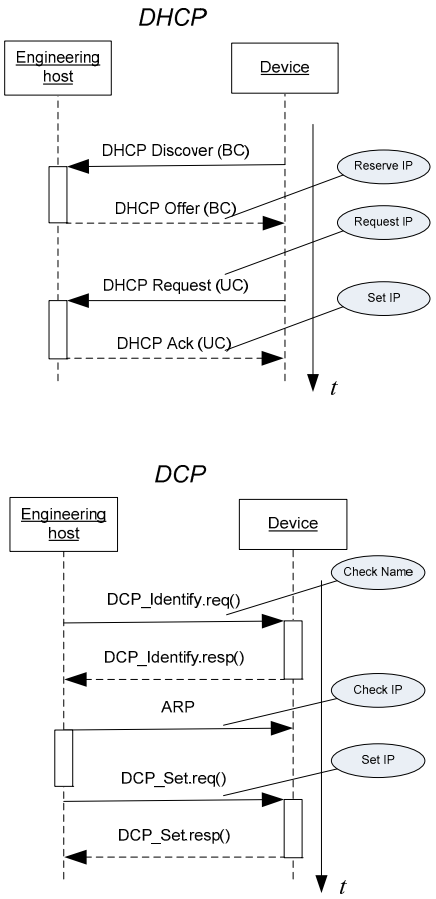

Fig. 2. Comparison of DHCP vs. DCP

PROFINET (PN) as standardized in IEC 61158 und IEC 61784-2 defines two methods for allocation of parameters like the IP address, subnet mask, default gateway and station name. PROFINET Class B devices must implement the Discovery and Configuration Protocol (DCP), DHCP is only optional. Hence, DHCP is not even implemented in most PROFINET stacks and address assignment tasks are entirely performed with DCP, which is a data link layer based protocol, operating with the PN multicast MAC address and Ethertype 0x8892.

In the discovery phase, the PN-Controller sends Identify-Request messages to the PN multicast MAC address. The PN-Devices answer with Identify-Response messages containing their MAC address, a device identifier and PN device name (if available). Based on these answers the PN-Controller can recognize new devices in the network and decide to assign a parameter set automatically. The same mechanism can of course be used manually, triggered by human interaction to detect devices in the network and assign IP addresses manually. The main difference is that with DCP the controller initiates communication, whereas with DHCP the device initiates communication by issuing DHCP-Discovery messages as broadcast (BC) followed by DHCP-Request messages as unicast (UC). The main differences of DHCP and DCP are shown in Fig. 2 by means of sequence diagrams of the assignment.

A topology based parameter assignment approach is realized within the PROFINET protocol suite by making use

of the “Link Layer Discovery Protocol” (LLDP) according to IEEE802.1AB in combination with DCP for configuration deployment. In [12], something similar is proposed, consisting of a ”rollin” procedure for discovering the network topology by means of knowing the end nodes and branches positions and a ”rollout” procedure, which is used to initialize the topology addressing of all nodes.

Hirschmann’s HiDiscovery protocol [13] is multicast MAC based and in fact an early version of DCP, which was implemented by Hirschmann during the specification phase, but later abandoned by the PNO working group and thus no longer compatible to the standardized version of DCP.

Other device vendors have developed similar protocols that use Ethernet multicasts or broadcasts to scan for devices in the network and remotely perform static IP address assignment. The disadvantages of such data link layer based protocols like DCP or HiDiscovery are the same as with RARP. Access to the lower layers of the communication stack is required on the computers that run the tool software for the parameter assignment. For modern operating systems this requires installation of additional software packages like WinPcap [14] to operate with messages on the data link layer. These protocols are also not routable and limited to be used in the same network segment.

To solve this, several other approaches have been developed by different vendors. Similar to BOOTP and DHCP, they are based on UDP-Broadcasts and dedicated UDP ports. This has the advantage that no data link layer access is required for the host computer operating systems[2]. Usually all of these approaches are provided each with dedicated PC based tools that have user interfaces similar to the Nut/OS Discoverer (cf. [15]), where the discovered network devices are listed with some parameters, which can be set via the protocol.

IV. Approach for a DHCP Extension for Networks without OPC UA

As already stated, the major obstacle for good usability of BOOTP and DHCP in industrial automation networks is the fact that communication can only be initiated from client devices towards server hosts. Approaches described in the previous section like DCP follow the opposite scheme, i.e. assignments are initiated by engineering hosts. This is a specific need for automation networks to allow scanning the network, manual configuration and dedicated configuration rollouts. In fact, a combination of both device- and host-initiated approaches would be required.

The FORCERENEW mechanism already exists for DHCP [16] and enables a host-initiated reconfiguration. It leads to a seven message exchange for the reconfiguration of a single client device and creates a lot of overhead. Since only client devices already known to the server can be reconfigured, unknown silent devices cannot be handled. The FORCERENEW process can also not deal with a DHCP server losing its lease database, i.e., due to a crash. There is no mechanism provided to easily recover the lease database i.e. by scanning the network.

To overcome these shortcomings, the DHCP Discovery Extensions are proposed to add host-initiated parameter assignment capability to DHCP. With this approach, the network can be scanned for devices in the network before allocating any IP addresses and this information can be used to maintain a lease database on the host. Duplicate IP addresses can be more easily detected and the administrator can perform dedicated reconfigurations on certain devices in the network. Thus, the DHCP Discovery Extensions allow flexible centralized network administration. It is also possible to adopt this approach for DHCPv6.

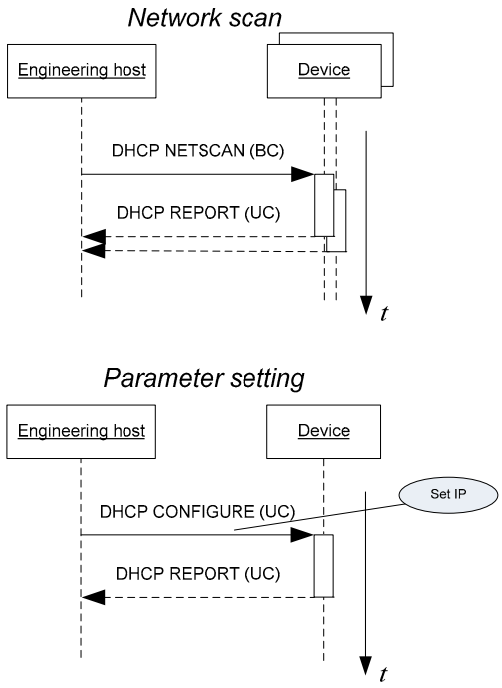

Fig. 3. DHCP Discovery Extension procedures

The DHCP Extensions propose three new DHCP messages as shown in Fig. 3. The first is for a network scanning, called DHCPNETSCAN. The host uses this message to request device parameter settings. The second message, DHCPCONFIGURE, is meant for device configuration with new parameter settings. The third message, DHCPREPORT, provides an acknowledgement mechanism and is sent as a response to network scanning and configuration messages. The report message contains the device’s current parameter settings. The DHCP Extensions also introduce the Discovery Option Flag, to remotely configure and report the administrative state of the extension functionality on the device.

This DHCP-based approach offers a good solution for the stated problem by extension of the widely accepted standard protocol DHCP. It would be perfectly suited to be considered for a unified IP address assignment approach whenever no OPC UA infrastructure is available.

[2] Hilscher NetIdent, www.hilscher.com
Westermo IP config protocol, www.westermo.com
Winford ETH32 UDP conf., www.winford.com
Turck IP Address Tool, www.turck.com
Comtrol Port Vision DX, www.comtrol.com

V. AUTO-CONFIGURATION AS OPC UNIFIED ARCHITECTURE EXTENSION

OPC UA is considered as platform-independent industrial middleware technology, and has recently gained a lot of attention as promising protocol technology for the Industry 4.0 reference architecture being currently specified. In this section an alternative approach for IP address assignment is presented, which is based on the existing OPC UA discovery mechanisms.

A. OPC UA Background

OPC UA is designed to allow interoperability between heterogeneous system components over various types of networks. OPC UA defines methods for both data modelling and transport. The information is modelled in the OPC UA address space by using object-oriented techniques. Since the OPC UA specification provides only an infrastructure to model information, it does not dictate any particular semantic. It is open to domain specific organizations to define their own respective information models. Thus, OPC UA is envisioned as an enabler for a seamless vertical integration in automation systems. The OPC UA architecture follows the server/client paradigm. A server can contain several endpoints, each offering information modeled in the endpoints' address space. A client can browse the address space and request the relevant data.

B. OPC UA Local Discovery and Global Discovery

OPC UA specification consists of currently thirteen parts, where part 12 "Discovery", while still in draft state, is of specific interest in this context. It specifies a set of discovery services by which OPC UA clients can gather information about OPC UA servers, including endpoint and security information. In the global discovery mode each server registers itself at a global discovery server – whose address must be pre-configured in all OPC UA devices. Such manual configuration is not necessary when using the local discovery service. Its limitation to local subnets is not a real constraint in automation systems like the SFOWL consisting of one local network. The OPC UA's local discovery service is using Multicast DNS (mDNS) as specified in RFC 6762 for device discovery, the typical sequence is shown in Fig. 4.

Each OPC UA host contains a local discovery server with multicast extensions (LDS-ME). When the OPC UA client needs information about the available servers, it sends a FindServersOnNetwork command to its local discovery server – which, in turn, sends a probe message to the mDNS multicast address. The server's LDS answers with an announce message containing the DiscoveryUrl of the OPC UA server. By using the DiscoveryUrl the OPC UA client can establish a direct connection to the OPC UA server. Technically, the multicast messages are using the DNS Service record (SRV record) syntax. In general, SRV records are used for requesting and announcing network services. A part of the SRV record is the service field indicating the requested or announced service. For example, OPC UA local discovery defines the service _opcua-tcp for LDS supporting TCP. The DiscoveryUrl is submitted in an additional DNS Text (TXT) Resource Record.

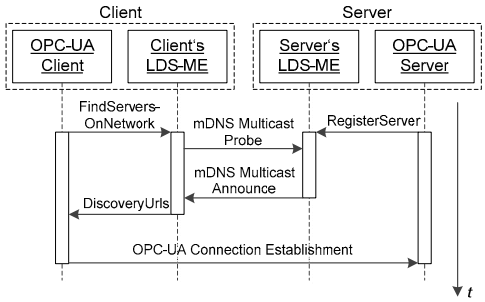

Fig. 4. OPC UA local discovery

C. Integration of the IP address assignment

Based on the local discovery service an IP address assignment method can be implemented in OPC UA which integrates the missing engineering host-initiated device configuration capability of DHCP. Furthermore, by using OPC UA for the address assignment the use of vendor-specific tools for device management will no longer be necessary.

In the following no distinction is made between OPC UA server and client. According to Fig. 2 the terms device and engineering host are used. Both the device and the engineering host include a LDS-ME for multicast operations. For the address assignment a new OPC UA SRV record service is defined, for example _opcua-devicemanagement. The type of operation, for example IP-Request or IP-Set, can be defined in a TXT Resource Record.

As described in the requirements in Sec. I the basic functionality of the address assignment method must cover two use cases, (i) device initiated assignment, and (ii) engineering-host initiated assignment. The approach for both use-cases is as follows.

1. Device-initiated address assignment (as in DHCP)

For the first use case, the IP assignment can be realized similar to the OPC UA local discovery process. The procedure is shown in Fig. 5. The multicast IP request of the device is sent to the mDNS multicast address and contains a request for the service _opcua-devicemanagement in its SRV record. After receiving the request the engineering host sends an answer using the same SRV record. The IP address is included in the TXT resource record.

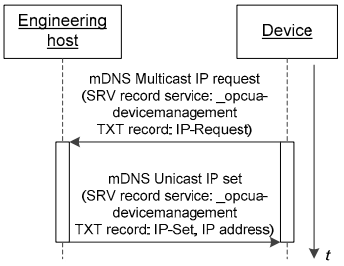

Fig. 5. Proposal for a devive-initiated address assignment in OPC UA

Further operations like the DHCP FORCERENEW command or the suggested DHCP extensions are not necessary here since, unlike as in DHCP, the engineering host is able to establish a connection to the device by itself.

2. *Engineering host-initiated address assignment (as in DCP)*

In this scenario the engineering host must be able to discover the available devices in the network (like DCP's identify request does) and to allocate IP addresses to them. The discovery process can again be built upon an mDNS multicast request using the SRV record _opcua-devicemanagement. Therefore a multicast identify request is sent by the engineering host, by using the TXT resource record it specifies if it wants to discovery all devices or only a certain subset, identified by a selector, for example their vendor-id, product-id or device name. The according devices send identify responses to the host. At last, the host can allocate the IP addresses to the devices as in the first use case. The procedure is shown in Fig. 6.

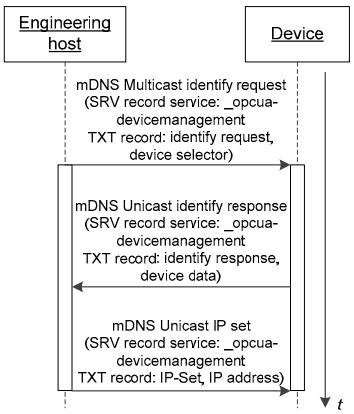

Fig. 6. Proposal for a host-initiated address assignment in OPC UA

VI. Conclusion

We highlighted the lack of commonly accepted unified mechanisms for initial IP address assignment in industrial networks. Solutions from the IT networks like DHCP miss important features required for industrial networks, such as proactive network scanning, manual IP address configuration and dedicated configuration rollouts. Industrial communication suites like Ethernet/IP and OPC UA yet do not define a mechanism at all. Consequently, we made two proposals for creating a standard solution. First an extension to Standard-DHCP for networks without OPC UA, and second an extension for OPC UA. Both mechanisms are based on similar communication mechanisms but define each their own message format. The DHCP Extensions would be advantageous, because it builds on established structures and the concept of the Relay Agent can be utilized to communicate across network segments and allow topology based IP address assignment policies. Also the implementation efforts might be significantly lower, because existing DHCP Client and Server implementations can be easily reused and extended. We therefore recommend the described DHCP-based mechanisms for OPC UA within specification part 12 "Discovery". In opposite to a complete new approach for OPC UA, this would very much ease the integration and migration of legacy systems into the new OPC UA world.

Future work will address the integration of the approach into the SFOWL as a first proof-of-concept within a real industrial system, followed by a thorough evaluation and possible optimizations of the approach.

Acknowledgment

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (it's OWL).

References

[1] L. Dürkop, H. Trsek, J. Jasperneite, L. Wisniewski. "Towards Autoconfiguration of Industrial Automation Systems: A Case Study using Profinet IO," *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on,* Sep. 2012.

[2] Internet Engineering Task Force. RFC's available at: https://tools.ietf.org/html/

[3] Gebremichael, Biniam, Frits Vaandrager, and Miaomiao Zhang. "Analysis of a protocol for dynamic configuration of IPv4 link local addresses using Uppaal," *ICIS, Radboud University Nijmegen, Tech. Rep. ICIS-R06xxx*, 2006.

[4] E. Guttman. "Autoconfiguration for IP networking: enabling local communication," *Internet Computing, IEEE* , vol.5, no.3, pp.81,86, May/June 2001.

[5] A. Swales. "IP Address Assignment in Large Industrial Networks," *Technical Report Network Vision,* Nov. 2003.

[6] J. Norman. "Using DHCP to Minimize Equipment Setup Time," *Technical Report N-TRON Corp.,* Aug. 2009.

[7] K. Weniger. "PACMAN: passive autoconfiguration for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol.23, no.3, pp.507,519, March 2005.

[8] M. Al-Shurman, M.F. Al-Mistarihi, A. Qudaimat. "Network Address Assignment In Mobile Ad-Hoc networks," *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*, pp.287,294, Oct. 2010.

[9] CJ. Bernados. "Survey of IP address autoconfiguration mechanisms for MANETs," IETF Internet draft, available at: https://tools.ietf.org/html/draft-bernardos-manet-autoconf-survey-05. Last accessed: 30.01.2016.

[10] M. Rentschler. "Faulty device replacement for industrial networks," *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, pp.57,62, 25-27 July 2012.

[11] A. Tominga, et al. "Problems and Solutions of DHCP," In: *The 5th Annual Conference of the Internet Society (INET 95)*. 1995.

[12] J. Jasperneite, J. Imtiaz, M. Schumacher, K. Weber. "A Proposal for a Generic Real-time Ethernet System," *IEEE Transactions on Industrial Informatics,* 5(2):75–85, May 2009.

[13] Hirschmann Automation & Control. "HiDiscovery Protocol," available at:http://www.hirschmann.com/en/Hirschmann_Produkte/Industrial_Ethernet/Software/Software_Tools/HiDiscovery/index.phtml. last accessed: 02.024.2016.

[14] Riverbed Technology. "WinPcap - The industry-standard Windows packet capture library," available at: http://www.winpcap.org/

[15] Nut/OS Discoverer; available at: http://www.ethernut.de/nutwiki/Discovery_Service, Last accessed: 06.02.2016.

[16] Internet Engineering Task Force, "RFC 3203 - DHCP reconfigure extension," Available at: https://tools.ietf.org/html/rfc3203, Last accessed: 05.02.2016.

[17] J. Wu, J. Dong. "A Simple Service Discovery and Configuration Protocol for Embedded Devices," *Communication Technology, 2006. ICCT '06. International Conference on*, pp.1,3, 27-30 Nov. 2006

[18] L. Dürkop, J. Imtiaz, H. Trsek, L. Wisniewski, and J. Jasperneite, "Using OPC UA for the Autoconfiguration of Real-time Ethernet Systems," in *11th IEEE International Conference on Industrial Informatics (INDIN)*, July 2013.

[19] Mai Son, Myeong-Jae Yi. "Towards an efficient discovery services in OPC unified architecture," Advanced Intelligent Computing. Springer Berlin Heidelberg, 2012. S. 544-552.

[20] Mai Son, Myeong-Jae Yi. "OPC UA Discovery: A Study of Challenges and Perspective", Applied Mechanics and Materials, Vols 157-158, pp. 110-113, Feb. 2012.