



Configuration Solution for SDN-Based Networks Interacting with Industrial Applications

Thomas Kobzan, Immanuel Blöcher, Maximilian Hendel, Simon Althoff, Sebastian Schriegel and Jürgen Jasperneite

Abstract

Versatile production systems are a main concept of Industrie 4.0. In order to achieve the needed flexibility and changeability of those systems, the industrial network has to become more adaptable as well. The density of the interconnection between industrial applications increases as well as the interconnection of the industrial network management and these applications. In order to configure the industrial network appropriately, a network management has to be able to identify the requirements of the industrial applications regarding their needed network services and the quality of service. Furthermore, it can be enabled to find alternative configurations, if certain requirements cannot be met, even if it has to influence the industrial process itself.

T. Kobzan (✉) · S. Schriegel · J. Jasperneite
Fraunhofer IOSB-INA, Lemgo, Deutschland
e-mail: thomas.kobzan@iosb-ina.fraunhofer.de; sebastian.schriegel@iosb-ina.fraunhofer.de;
juergen.jasperneite@iosb-ina.fraunhofer.de; juergen.jasperneite@th-owl.de

I. Blöcher · M. Hendel
Hilscher Gesellschaft für Systemautomation mbH, Hattersheim, Deutschland
e-mail: ibloecher@hilscher.com; mhendel@hilscher.com

S. Althoff
Weidmüller GmbH & Co KG, Detmold, Deutschland
e-mail: simon.althoff@weidmueller.com

J. Jasperneite
Institut für Industrielle Informationstechnik inIT, Lemgo, Deutschland
e-mail: juergen.jasperneite@iosb-ina.fraunhofer.de; juergen.jasperneite@th-owl.de

This paper presents a solution that uses Software-defined Networking (SDN) and OPC UA in combination and achieves a higher amount of flexibility by interacting with the industrial process. For this, the network management gets enabled to influence the industrial process if the quality of service demands of the industrial applications cannot be guaranteed. Additionally, it is able to switch network paths to alternative routes and configures them by using OpenFlow-based SDN switches. The solution is implemented according to a described industrial use case, which underlines the necessity of this solution.

Keywords

Industrial internet of things · Software-defined networking · OPC UA

1 Introduction

Industrie 4.0 promotes the coming of highly versatile production systems. These systems are not only flexible, meaning adapting within a predefined range, but they are also changeable [1]. Changeability is the skill of a production system to be reassembled by adding or removing modules with the goal to quickly get enabled to produce totally different products. This capability is not only demanding for deterministic processes, but also for the performance of data-driven services like data analytics or machine optimization. Because this data must travel from its generation location, usually sensors in the production field, to its destination for processing, usually some kind of server in some remote location. An ever-changing production environment requires an industrial network that does not only guarantee that the data is able to travel from source to destination, but also with the required quality of service (QoS). Because of the complexity, it occurs that the QoS cannot be met from time to time.

Industrie 4.0 tries to solve these tasks using appropriate functionalities. A special focus lies on the configuration part of the industrial network. Recently, the networking paradigm Software-defined networking (SDN) and the communication technology OPC UA are under special surveillance for solving these tasks [2, 3, 4]. SDN is in discussion as a solution for the flexible and dynamic configuration of the networks by (re)programming paths for the appropriate communication between the network participants. Furthermore, there is OPC UA as a potent communication technology, which is not only useful for the communication among the network participants, but also for the interaction between them and the management entities of the industrial network. This interaction does not have to stop at the mere exchange of demands on the QoS and other communication-related requirements. In order to guarantee that the demands on the QoS are always met, a centralized software-based network management could be enabled to execute functionalities beyond that. Prerequisite is that the network management is able to interact with the industrial applications.

This brings up the question, how such an interaction can look like when using programmable networks based on SDN in combination with OPC UA and what possibilities this offers for future industrial networks.

The rest of the paper is structured as follows: For a better understanding, Sect. 2 gives an example in form of an industrial use case. Section 3 describes the basics of the technologies and Sect. 4 gives an overview of the related work. Section 5 describes the concept of the architecture, which is the basis for a first implementation depicted in Sect. 6. Section 7 discusses the results. Section 8 concludes the paper and gives an overview of potential new research activities.

2 Industrial Use Case

In a production environment, the production quality can be increased by conducting visual inspection on certain products. Therefore, a product, which is conveyed with a high speed, is detected by a sensor sending a trigger signal to a camera, which acquires the image. The trigger data is sent through the network, so it must be transferred with a certain determinism. If the QoS is low, the trigger signal may be received too late or is provided with a high jitter, what means the product is not in the perfect spot for the image detection. This lowers the quality of the image exploitation or makes it impossible. In an Industrie 4.0 scenario, the industrial network may be empowered with certain functionalities to guarantee the QoS. These functionalities can go beyond the current capabilities, because of the tighter interconnection of the components (dissolving of the automation pyramid and convergence of IT and OT to Industrial Internet of Things [5]) and the usage of services for network optimization.

Figure 1 shows the simplified use case. The devices whose interaction needs deterministic treatment are the Object Detection and the Visual Inspection. Both have an access to

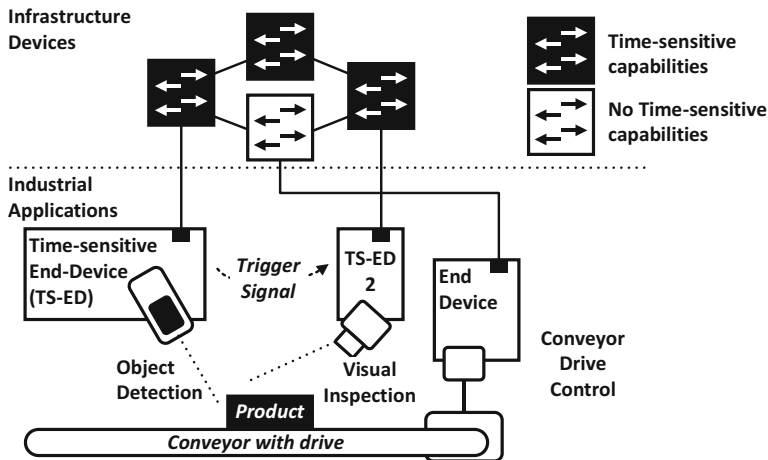


Fig. 1 Industrial use case describing the need for flexible network configuration and its potential

the industrial network, which is shown in a simplified fashion as well. The pure transmission of the trigger signal can be processed via the route with pure time-sensitive switches or via a route with a switch without time-sensitive capabilities in between. This offers the possibility for the industrial network to send the signal using the time-sensitive path. This may be mandatory, if the product moves with a high speed and there are demanding constraints regarding the quality of the image acquisition. If not, it may be sufficient to transfer the signal via the switch without time-sensitive capabilities and save time-sensitive resources. An additional alternative is the slowing-down of the conveyor. As it can be seen, this requires a tight interconnection of a networking management entity with the industrial applications and the entities managing the production process. Thus, the network is able to gather all information it needs to apply the most appropriate network configuration and adjust the industrial process if needed. Furthermore, it requires a fair amount of network optimization methods that help the network to find these suitable configurations.

3 Basics

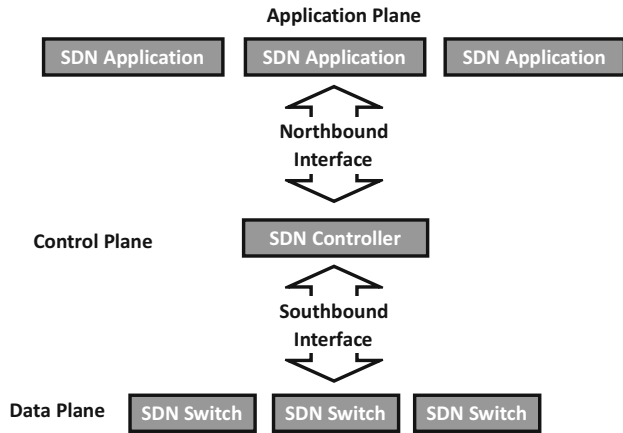
This section describes the basics of the main technologies and the benefit from combining them.

3.1 Software-Defined Networking

Software-defined networking (SDN) is a networking paradigm where the control of the forwarding instructions is located in a central entity. Conventional network infrastructure devices like switches or routers include this control functionality. The data plane, where the forwarding occurs, and the control plane, where the decision how to forward occurs, are located on the same device. By separating these functionalities, SDN is said to be more flexible, because the central control entity is able to react to the network demands on its own and by programming the forwarding decisions onto the controlled infrastructure devices.

Figure 2 shows a simplified structure of the SDN paradigm. As mentioned, the control plane is separated, so the SDN controller is located on its own control plane. The control plane is connected via a Southbound interface to the data plane, where the forwarding instructions are executed. On this data plane, the SDN switches are located. It is up to the individual implementation to which extent the SDN switches are able to forward data. In the dominant implementation called OpenFlow [6] the switches are even able to identify information above OSI layer 2 in the datagrams and execute forwarding instructions based on the programmed instructions. Above the control plane, the application plane is located, which connects to the control layer via a Northbound interface. The SDN applications expand the functionality of the network by implementing network functions and influencing the controller for the correct execution. Example usages for these programs

Fig. 2 Software-defined networking architecture



are intrusion detection and network monitoring, which are often implemented as virtual network functions.

3.2 OPC UA

OPC UA (Open Platform Communications Unified Architecture) is an open cross-platform technology for the communication between machines. There are two main communication models: server-client and publisher-subscriber (PubSub) [7]. In the server-client model, a client connects to the server and acquires or is able to edit the reachable data. When there are many nodes in a network interconnecting, this form of connection results in heavy network load. To overcome this, the PubSub model was implemented, which relates to different transport protocols like, e.g., MQTT or UDP. A publisher may send its data to a central entity called a broker. Subscribers can connect to this broker browsing and asking for relevant published data. There exist brokerless variants as well, where the publishers' messages reach the subscribers without utilizing a broker.

A main advantage of OPC UA is the feature for the implementation of own information models. Moreover, OPC UA natively implements possibilities for security like authentication and authorization.

3.3 Combined Usage

As a machine-to-machine communication technology, OPC UA can be utilized by SDN, more precisely, by the controlling or managing entities of a software-defined network. As mentioned, centralized entities are utilized in SDN for configuring and instructing how packets are forwarded in the network. These centralized entities can make their decisions

upon the demands of the network participants. These participants are normally industrial applications that need to fulfil certain services in an industrial production.

OPC UA can be utilized as a communication tool between the industrial applications and the centralized entities. The centralized managing entities can get the information of the application requirements in order to apply the network services the industrial applications need. Beyond that, the network may get empowered to understand the industrial production process and, therefore, can optimize its service for the industrial applications even more. For this, OPC UA can not only be used for the gathering of pure information but also for altering data on the devices by the industrial network.

4 Related Work

This section provides an overview over relevant work. The authors in [8, 9] discuss the relevance of Software-defined networking as a potential technology for flexibilizing industrial networks. They describe the possibility to see the industrial communication respectively the industrial network as Industrie 4.0 component extended by an Asset Administration Shell according to the reference architecture model industrie 4.0 (RAMI 4.0). OPC UA may be a candidate to implement the information models that are needed for this. The usage of OPC UA is proposed by the authors in [10] as well. They conclude among other things that uniform automation models form the basis for the flexible industrial networks of the future and these networks need self-adaption during runtime. These adaptable networks may be based, amongst others, on SDN.

Madiwalar et al. [4] described in 2019 a solution for Plug and Produce using SDN and OPC UA. They demonstrated that it is possible to use these technologies for a device's registration process with the coordinator device. Gerhard et al. [3] proposed a solution for configuring time-sensitive networks using SDN and OPC UA. They used SDN and a well-established SDN-controller platform for the configuration of time-sensitive data-paths. They used OPC UA as a communication technology between network participants and the network management to transmit demands and perform auto-configuration on the data plane according to the information gathered from the participants. Bruckner et al. [7] describe the potential of OPC UA to be used as data plane communication between machines combining it with TSN. For the configuration of such networks, they describe some kind of central configuration manager that has decent knowledge of the network, e.g. about topology and the devices configuration. They state that the configuration is getting more complex if it has to deal with bridged endpoints that include the functionalities of switches and endpoints. Ansah et al. [11] described in 2020 a solution named a controller of controllers architecture. They pursue a fully centralized network management approach where centralized entity has the overview of the whole network and manages it over a plurality of network controllers that each use a specific network technology like TSN, SDN, PROFINET.

5 Architecture

This section describes a simplified configuration architecture with the help of Fig. 3. More elaborate architecture concepts are described in [11], but this section shall help to get a better understanding, how the network can be configured by means of the interactions of applications, and is the basis for the implementation of Sect. 6.

The Network Data Layer here represents the plane, where the data exchange between the industrial devices occurs. Every device that generates, consumes, or forwards these data is located here. This applies for infrastructure devices as well as for devices that belong to an industrial application. The only differentiation between these devices is the amount of application-related or communication-related functionality. A Smart Device like the smart camera mentioned in Sect. 2 has functionality in both areas. Its application for the industrial process is the acquisition and exploitation of images. For the communication, it includes a network interface that may be able to communicate over an industrial protocol and thus, may be able to transmit or receive time-sensitive data. Another example are bridged endpoints. These devices typically are part of an industrial application, but they include as well an embedded switch, where one port is connected internally and two ports are for external connectivity. Therefore, they are designed in order to form, e.g., line topologies typically found in industrial networks. Common infrastructure equipment like switches or devices with routing functionality may have little to no application-related fraction or, as in the example with the bridged endpoints, they form the communication-related fraction in a more complex device respectively application. A complex production module can present itself as sub-network that has one dedicated entry

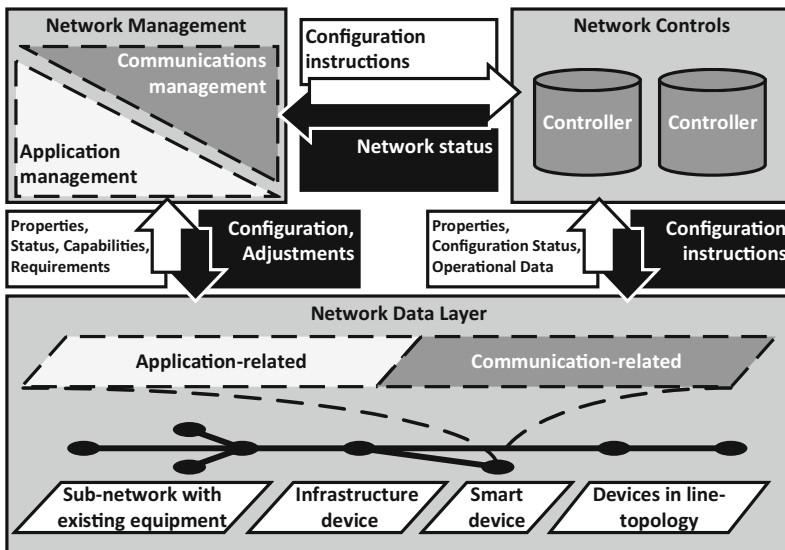


Fig. 3 Simplified architecture

point for the external communication and may include a plurality of application-related functionality.

Regarding the configuration of the devices located on the network data layer, the whole architecture follows a centralized approach very similar to the SDN paradigm or the fully-centralized configuration model that is defined in the IEEE 802.1Qcc-2018 amendment specified by the Time-sensitive Networking Task Group. The communication-related fraction is managed over network controllers that are located in a network controls layer. The need for more than one controller comes on the one hand from the still present heterogeneity of industrial ethernet technologies and on the other hand from the need to separate the network into functional parts.

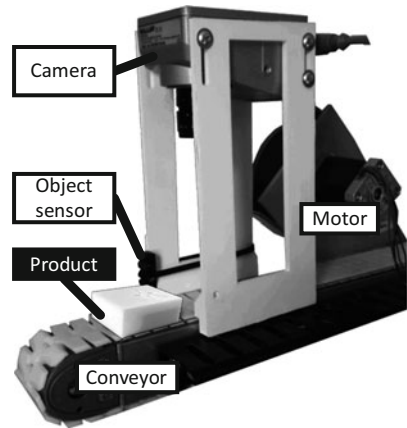
The network management represents here the sole centralized managing entity that gets the network status from the controllers and gives them instructions on how to configure the network. It also has a link to the network data layer and can get properties, status, capabilities and requirements of the devices respectively applications here. This does not stop at the application-related functionalities. For adequate configuration decisions, it may be additionally helpful whether the network management knows communication-related information like, e.g., the capabilities of a switch that is able to transfer data with certain time-sensitive properties. If this information cannot be received through the controller, it may be helpful, if the network management can get this information directly.

The simplicity of this architecture comes with a plurality of tasks to be solved. This paper deals especially with the interfaces between network data layer, controls and management. The process how all the information gathered by the management can be transformed in an adequate and reasonable configuration is out of scope, but a simplified example is given in the implementation section. It is as well out of scope, how the plurality of technologies can be managed in the future. SDN and the dominant management protocol OpenFlow are an adequate solution for a programmable network that can deal with the configuration that results from the interaction of the applications. OPC UA is a suitable communication technology for transferring information from the network data layer to the network management as well as receiving information or configuration instructions from it. Additionally, OPC UA is a proven technology for implementing data models.

6 Implementation

Based on the architecture of Sect. 5, this section describes a first approach to implement the use case of Sect. 2. The demonstrator is depicted in Fig. 4. The image only shows the devices that form the industrial applications. The conveyor moves the product, what is a 3D printed block in the figure. The product is detected by the object (laser) sensor. The sensor constantly sends its detection status to the trigger input of the smart camera. The trigger is activated when the product leaves the detection area, and an image is acquired. The sensor is an optoelectronic sensor (BOS R020K-PS-RH12-00,2-S75) and the camera is a smart camera (BVS002A – BVS SC-M1280Z00-30-000), both from BALLUFF. As

Fig. 4 Demonstrator: 3D printed block is conveyed and triggers the camera by leaving the detection area of the object sensor



mentioned in Sect. 2, the sensor output and the trigger input are not directly connected. They are connected to evaluation boards from Hilscher which feature a netX 90 SoC. These evaluation boards serve as the industrial devices that feature application-related and communication-related functionality. In combination with the sensor device and the smart camera (trigger input), they form an industrial application by means of the architecture of Sect. 5. They include an information model, implemented through an OPC-UA-server, that features the information, what they do in the context of the industrial process (application) and how they are able to exchange data (communication). They are able to transfer the data using methods described in IEEE 802.1Q-2018 and IEEE 802.1AS-2011. The specific methods here are the commonly known Time Aware Shaper and the gPTP synchronization. Therefore, they are able to use TSN as time-sensitive technology for transferring the trigger signal from the sensor to the trigger input of the camera. The data is encapsulated as an OPC UA PubSub frame.

A third device from Hilscher without TSN functionality is additionally implemented. A Hilscher netPi is utilized to control the motor driver of the conveyor. It also hosts an information model in form of an OPC-UA-server. Via this model, the speed of the conveyor is accessible and adjustable.

6.1 Topology and Network Configuration

The topology and the information exchange mechanism for the network configuration is depicted in Fig. 5. The programmable network consists of four Northbound networks “Zodiac FX” SDN switches. The management protocol is OpenFlow 1.3 and the controller is OpenDaylight (ODL) 0.8.4. This first approach features only one controller which runs on the same hardware device as the management functions that connect the devices. Therefore, the southbound connection between SDN switches and controller is OpenFlow

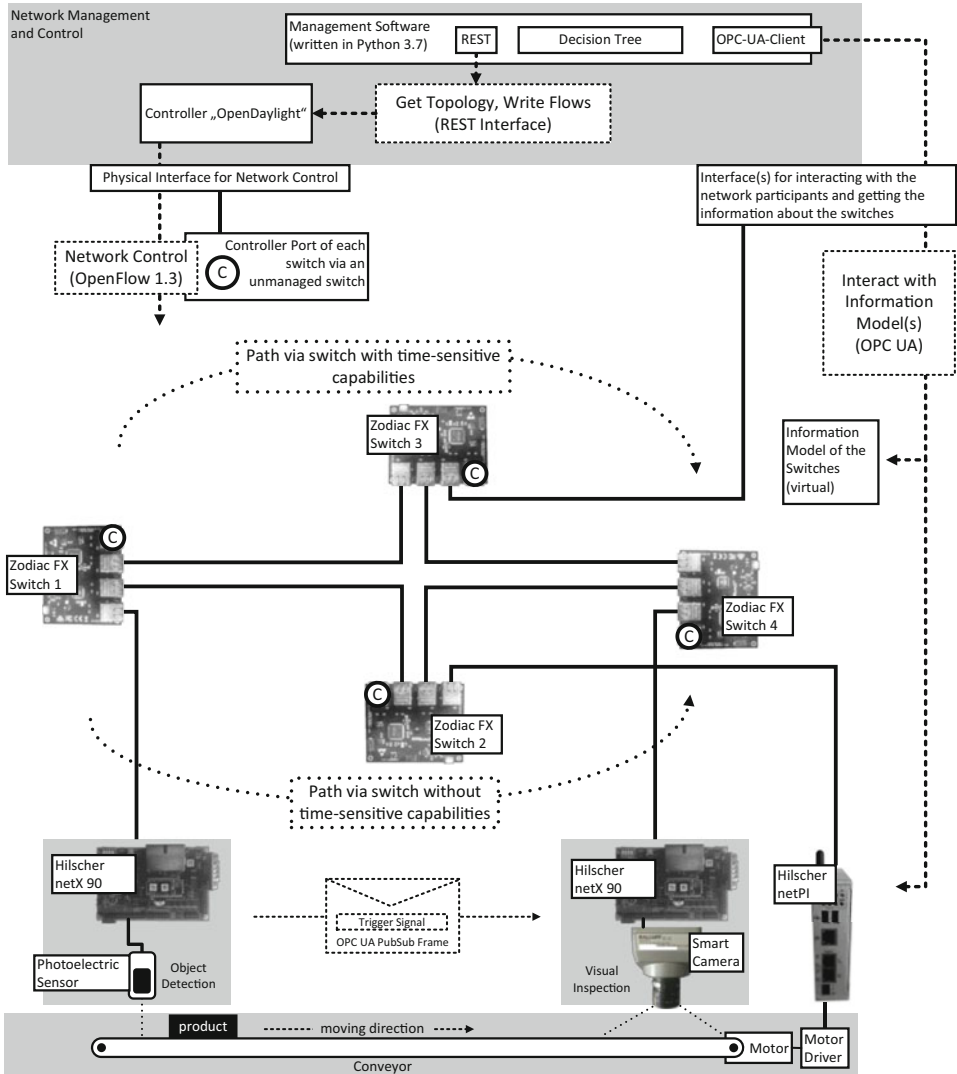


Fig. 5 Topology and information exchange structure of the implementation

1.3. The connection between controller and management is done via the REST interface of the ODL controller. The network programming routines of the management are established in Python 3.7. In the default configuration, the ODL controller configures the network in a way that all switches act like unmanaged switches. This configuration was replaced by a more restrictive initial configuration. Only the communication of the devices (netX boards and netPI) with the management is possible. The communication amongst the devices is not permitted. The forwarding of LLDP and ARP is not restricted, because this is necessary

for the generation of the topology by the network controller. The topology is a mandatory information for the management.

The network management gets the topology from the controller and the information models from the applications. Additionally, information models for the switches were implemented in a remote virtual environment. These information models include the information about the TSN capability of the switches. It must be noted that this is hypothetical, because the Zodiac switches do not include TSN functionality.

6.2 Configuration Example

In order to show the use case described in Sect. 2, Fig. 6 shows a decision tree, which shall demonstrate a simplified deduction of observable parameters into configuration decisions. The start “request for transmission of trigger signal” means, the program receives the order to send data from a source (object sensor evaluation board) to a destination (smart camera evaluation board). The roles of the network participants are defined and the network “knows” these roles. A semantic deduction or other kinds of logic to achieve a knowledge of these roles is out of scope.

The network management reads the value of the conveyor speed from the information model of the netPI. Based on this, it decides whether it is mandatory to install a time-sensitive path or not. If not, it saves resources on the time-sensitive capable switch 3 and configures a path over switch 2. If it is mandatory, it checks whether time-sensitive resources are available. This is done in this example by observing the traffic on the port of switch 3, which is connected to switch 4. The traffic is increased by using the tool Ostinato (ver. 0.8) and sending traffic from the network management connection at switch 3 in the direction of switch 4. Normally, this could be also done by checking the configured TSN parameters like the schedule of a set Time Aware Shaper and how occupied the priority

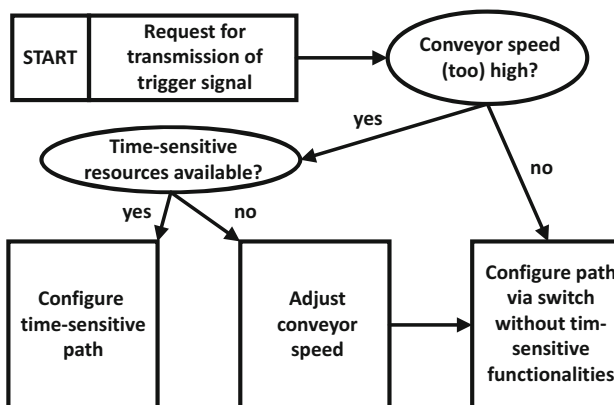


Fig. 6 Example for deducing a network configuration

queues are. If resources are available, a path via switch 3 is configured. If not, the conveyor speed is reduced by the network management and the path via switch 2 is configured. All decisions are predefined, the values are hypothetical. The path from source to destination is calculated using a Dijkstra algorithm. If the decision tree concludes that a path via switch 2 has to be configured, switch 3 gets blacklisted, leaving this path as the only one available. If the decision tree concludes to configure the time-sensitive path, switch 3 gets chosen, because all connections directed from time-sensitive capable switches have applied a lower distance value used by the Dijkstra algorithm. Therefore, the cost via the non-time-sensitive switch is higher and the other path gets chosen.

7 Discussion

The implementation uses OpenFlow 1.3 capable infrastructure devices with no time-sensitive capabilities like TSN. Depending on the TSN methods, an additional solution for the configuration would be to put the traffic in a proper queue of a Time Aware Shaper integration.

On the data plane, the solution is only relevant when programmable networks are used. Because the software is written in Python and OpenFlow 1.3 is used in combination with the OpenDaylight controller, it is capable of being integrated in the results described in [3] and [12], where solutions based on the combination of TSN and SDN are pursued.

Moreover, Python is usable for the integration of data-driven analytics for elaborating the decision finding. For this, more complex networks with more parameters to observe and adjust must be defined.

In a normal production process, it must be checked, if it is allowed to slow down the conveyor. This could endanger subsequent production steps or put the supply chain in jeopardy. Therefore, the industrial network management must not only interact with the industrial applications, but with the manufacturing execution system(s) as well.

8 Conclusion

This paper describes how the configuration of programmable networks may be enhanced by the interaction between the industrial applications themselves and the interaction between them and the industrial network. The described use case results in a simplified example. However, it illustrates the solution space that broadens, if the industrial network understands the interaction of the applications and can interact with the applications itself to optimize the network configuration and find appropriate solutions that go beyond the current possibilities.

The amount of interaction possibilities that Industrie 4.0 respectively the Industrial Internet of Things will demand for data-driven methods that are able to properly decide on the appropriate configuration of the network. Future activities may investigate such

methods. An additional research field is the definition, to which extend and how exactly the industrial network should be allowed to interact respectively interfere with the industrial process.

References

1. Malakuti, S., Bock, J., Weser, M., Venet, P., Zimmermann, P., Wiegand, M., Grothoff, J., Wagner, C., Bayha, A.: Challenges in Skill-based Engineering of Industrial Automation Systems. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin (2018).
2. Wollschläger, M., Sauter, T., Jasperneite, J.: The future of industrial communication. In: IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17–27 (2017).
3. Gerhard, T., Kobzan, T., Blöcher, I., Hendel, M.: Software-defined Flow Reservation: Configuring IEEE 802.1Q Time-Sensitive Networks by the Use of Software-Defined Networking. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza (2019).
4. Madiwalar B., Schneider, B., Profanter, S.: Plug and Produce for Industry 4.0 using Software-defined Networking and OPC UA. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza (2019).
5. Felser M., Rentschler M., Kleinberg O. Coexistence Standardization of Operation Technology and Information Technology. In: Proceedings of the IEEE, vol. 107, iss. 6, pp. 962–976 (2019).
6. Cox, J. H., Chung, J., Donovan, S., Ivey, J., Clark, R. J., Riley, G., Owen, H. L. Advancing software-defined networks: A survey. In: IEEE Access, vol. 5, pp. 25487–25526 (2017).
7. Bruckner, D., Stănică, M., Blair, R., Schriegel, S., Kehrer, S., Seewald, M., Sauter, T.: An introduction to OPC UA TSN for industrial communication systems. In: Proceedings of the IEEE, vol. 107, no. 6, pp. 1121–1131 (2019).
8. Rauchhaupt, L., Kadel, G., Mildner, F., Kärcher, B., Heidel, R., Graf, U., Schewe, F., Schulz, D.: Network-based Communication for Industrie 4.0 – Proposal for an Administration Shell. In: Plattform Industrie 4.0, Federal Ministry for Economic Affairs and Energy (BMWi) (2016)
9. Wollschläger, M., Debes, T., Kalhoff, J., Wickinger, J., Dietz, H., Feldmeier, G., Michels, J. Scholing, H., Billmann, M.: Communication in the Context of Industrie 4.0. In: ZVEI – German Electrical and Electronic Manufacturers’ Association (2019).
10. Jasperneite, J., Sauter, T., Wollschläger, M.: Why We Need Automation Models: Handling Complexity in Industry 4.0 and the Internet of Things. In: IEEE Industrial Electronics Magazine, vol. 14, no. 1, pp. 29–40 (2020).
11. Ansah, F., Soler Perez Olaya, S., Krummacker, D., Fischer, C., Winkel, A., Guillaume, R., Wisniewski, L., Ehrlich, M., Mandarawi, W., Trsek, H., de Meer, H., Wollschläger, M., Schotten, H. D., Jasperneite, J.: Controller of Controllers Architecture for Management of Heterogeneous Industrial Networks. In: 2020 16th IEEE International Conference on Factory Communication Systems (WFCS), Porto (2020).
12. Kobzan, T., Blöcher, I., Hendel, M., Althoff, S., Gerhard, A., Schriegel, S., Jasperneite, J.: Configuration Solution for TSN-based Industrial Networks utilizing SDN and OPC UA. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna (2020).

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

