Mesh Networking in Cyber-Physical Production Systems: Towards Modular Industrial Equipment Integration

Maxim Ya. Afanasev, Anastasiya A. Krylova, Sergey A. Shorokhov, Yuri V. Fedosov, Kseniia V. Zimenko
ITMO University
Saint-Petersburg, Russia
amax@niuitmo.ru, {ananasn94, stratumxspb}@gmail.com, yf01@yandex.ru, zksenia@yahoo.com

***Abstract*—Ensuring uninterrupted interaction of modular industrial equipment units is one of the most important engineering tasks. The concept of Cyber-Physical Production Systems (CPPS) assumes that the distributed network should correspond to the current industrial process and be able to quickly reorganize it when changes occur. If composition of the equipment becomes more complicated, a standard topology with one central control node might get ineffective. This article describes the application of mesh-network technology to ensure the interaction of industrial devices and sensors included in the modular equipment. Virtual deployment of the network and a description network nodes interaction including new node registration in the dispatcher registry are given.**

I. Introduction

Within the concepts of Cyber-Physical Production Systems (CPPS) modular architecture is getting more appropriate and gradually phasing out devices with a single monolithic architecture. Current trends require increased production flexibility in order to respond to the product range changes and customer requirements in time. It is assumed that modularity is not only a division into blocks in a physical way but also a modular software. In particular, the authors have already implemented the software architecture of modular equipment using the microservice approach [1]. This article brings up a question of creating a network infrastructure that uses modular equipment and special features of data transfer protocols.

Upon that, a division into unified blocks to simplify the architecture of industrial equipment leads to the inevitable complication of the connections between the equipment components located in a common decentralized network. However, at the moment there is no uniform standard for industrial distributed networks and the organization of network elements interaction among different companies is done in its own way.

On this basis, one of the main problems of existing CPPS is network implementation, which must meet the requirements of fault tolerance, security, and operation speed. The more production equipment is located on the production site, the more difficult to organize the interaction of network participants with each other. The reason behind this is a large number of connections central node is forced to work with within typical “star” network. In addition, the topology of devices and sensors distribution in the production area can be significantly different depending on the type of production process. Based on the foregoing, the network architecture being developed must have the capacity for self-organization and self-recovery.

One of the options for implementing the network architecture is a technology of mesh networks. Mesh network equipment allows choosing optimal frequencies and data transmission routes in automatic mode. The standard industrial network implies that each device is a closed unit of equipment consisting of interacting modules on the network, which are controlled by a special control module (dispatcher) [2].

Obviously, for this type of architecture, the classic “star” is more suitable. It allows quick subordinate control of each module without the need to search for it on the network and build an optimal route. Thus, each unit must stay closed and monolithic in order to solve directly the production tasks, because simple control model makes it more reliable and fault-tolerant. On the other hand, CPPS is not just a set of industrial equipment and sensors of the production environment. The second important entity of CPPS is a complete production model, also known as a “digital twin”.

The presence of a “digital twin” is the main difference between CPPS and ordinary industrial sensor or control network. The control effect on the model should lead to a change in the physical component of CPPS and vice versa. At the same time, the system as a whole should stay absolutely transparent to the user e. g., replacing the physical components of CPPS with virtual ones should not affect the system functions. Obviously, to implement such a concept, the model should be as detailed as possible. For that reason, it is proposed not to use a monolithic approach to the industrial equipment design but instead a modular one with the possibility of combining all CPPS components into one decentralized network.

This results in a variation of the so-called holonic approach when the same system can be both hierarchical and heterarchical depending on working conditions. It should be explained with the example of the designing system. As it was already noted earlier, each unit of industrial equipment is a set of modules controlled by a common dispatcher over a network. Each module is an autonomous entity with its own internal operation logic, actuators, sensors, etc. With a decentralized network architecture, data from each module comes not only to the direct dispatcher connected with it but directly to the CPPS model.

An abnormal situation or a failure of any of the equipment modules would be immediately translated through the network. With that, the “digital twin” based on its own logic can immediately decide on the possibility of continuing the operation of an equipment unit in the module where failure occurred (e. g., a failure occurred in an unresponsive or not involved at the moment module), or on the operational change of the production process and the transfer of the task to another (workable or unoccupied) unit of equipment. Additionally, this information will be distributed throughout the network (it should not be forgotten that CPPS is also integrated with cloud services), and this will work even if the failure occurs in the dispatcher module.

With this approach, the most reasonable way is to switch to the wireless organization of data transmission channels. In the case when the topology of the network is constantly changing, the installation of wire connections will be complex and expensive, besides there may be places with difficult access on the site. Wired connections will inevitably be damaged and rubbed in places of bends, which directly affects the overall reliability of the network infrastructure.

Summarizing all the above, the following requirements for the mesh network aimed to integrate modular equipment into CPPS can be formulated:

- The usage of the wireless method of data transfer.
- Openness and the possibility of integration with other production support systems: SCADA (Supervisory Control And Data Acquisition), cloud services, etc.
- High resiliency and the ability to self-repair.
- An embedded security system for data transmission.
- The usage of open standards.
- The network presentation as a digital production model, being the basis of the “digital twin”.
- The unnecessity of network nodes manual configuration (zero configuration principle).

The remainder of this paper is organized as follows. In Section 2 the previous work on wireless production networks is presented including examples of MANET and SDN technologies. Section III focuses on the current progress in wireless production networks area as well as on the existing standards and their pros and cons.

Section IV describes the proposed approach to implement a wireless network for modular equipment. Section V discusses the future work and the existing challenges which the proposed approach does not solve. The conclusion is provided in Section VI.

II. Related Work

Dealing with challenges in network infrastructure is of current interest, that is underpinned by gradual enabling of Industry 4.0 principles. Obviously, the future of production is behind wireless networks. Speaking of wireless production networks, first of all, coverage and interference issues are usually raised [3]. One of the solutions—mesh networks application. In [4] authors consider such a network, serving for interaction and management of mobile vehicles. They developed a combined approach, where an existing production network infrastructure and a mesh network are used together. Authors in their research [5] employ the combined approach as well. They suggest IEEE 802.15.4 channel based mesh network that employed for point to point communication, while broadcast traffic is transmitted by the sub-GHz band in one hop. This channel is intended to determine an emergency or the case of communication loss. However, we reject the idea of using a combined approach. In our case, an emergency has to be resolved at the level of modular equipment via a dispatcher. This means that interaction will be executed in one hop without any additional channels. Along with that, elimination of the emergency at the level of the production network can be carried out with some delay, since this will not lead to fatal consequences.

In addition to the above, production network managing and configuration is a hot topic of a discussion. One of the solutions to this problem is to use the so-called self-organizing ad-hoc or MANET networks, where no additional configuration is required when adding nodes to the network topology. In the study [6] authors are coming up with an ad-hoc sensor network for work under extreme conditions, such as power plants or warships. For these purposes, they develop OCARIN technology based on ZigBee standard. However, to meet harsh requirements, ZigBee was extended. Authors add deterministic MAC layer for time-constrained communication and a more sophisticated algorithm for saving energy, allowing to enter the energy-saving mode not only for end devices but also controllers. Moreover, unlike traditional sensor networks, where nodes are stationary, OCARIN supports mobile nodes. We are trying to contribute in ad-hoc networks as well and propose our own version of the organization of a mesh ad-hoc network. However, unlike the studies above, this article is focused on the integration of modular technological equipment, that introduces its own features into the architecture of the network and requires a separate consideration.

In parallel with ad-hoc networks, such technologies as SDN and SON [7] are getting known. This approach doesn’t allow one to achieve “zero configuration” principle but essentially simplifies network managing by making the controlled panel into a separate network node. The article [8] discusses its possible implementation. The authors note that in practice, many industries are already abandoning the use of fieldbus technologies, such as Modbus, Profibus, etc. in favor of industrial Ethernet. In such case, SDN application gives some benefits, e. g. remote update of PLC firmware and traffic management for interlevel communication between field and plant. The authors developed the architecture employing SDN technology and demonstrated a testbed based on such network and Raspberry Pi controllers.

Some authors are trying to combine the main principles of wireless ad-hoc networks and SDN to manage traffic manually. Thus, the authors of [9] research the possibility of creating a self-organizing network for tactical military purposes. The authors note that the application of general MANET is not enough, e.g. for coalition operations, where sensitive information has to circulate only between a certain group and never reach other network nodes. Thereby, the authors try to achieve benefits of centralized control as well as the ability to autonomously rebuild a network topology by enabling SDN technology. SDN obvious drawback is the centralized control and the presence of a single point of failure. Moreover, this technology, unlike the approach considered in this article, does not contribute to the principles of “zero configuration” and “plug and produce”, which are important for modular equipment and flexible re-configurable production.

The community is actively discussing the choice of the channel type for wireless data transmission in production. Currently, it is almost impossible to achieve control in a closed loop in a large production network due to delays and lack of reliability. However, with the advent of 5G networks, this will become realizable. There is an active discussion on the development of telecommunication standard 5G for production purposes [10]. The currently developed 5G standard will make it possible to get rid of all drawbacks mentioned above. In particular, Audi and Ericsson have recently launched a collaborative project [11] intended to introduce advanced communication systems based on 5G for industrial purposes that meet the requirements of the Industry 4.0. Moreover, the questions of integrating 5G networks with other types of networks existing on the plant are being considered, since it is obvious that 5g will not be able to displace them instantly [12]. However, in the current realities for the network described in this article, we focused on the architecture of the mesh based on the 802.15.4 channel and the OpenThread technology.

III. State-of-the-Art of Mesh Networking

Currently, there are four basic approaches to industrial mesh networks design:

- based on the set of standards IEEE 802.11 (Wi-Fi),
- based on the IEEE 802.15.1 standard (Bluetooth 4+, at usual Bluetooth Low Energy—BLE),
- based on cellular networks,
- based on the IEEE 802.15.4 standard (different variations of high-level protocols on the basis of special radio-channel with low bandwidth).

Each of the existing technologies is considered below in details.

A. *Mesh networks based on IEEE 802.11 standard*

The set of IEEE 802.11 standards defines the communication of devices in wireless computer networks in the frequency ranges from 900 MHz to 60 GHz. In practice, such networks are called Wi-Fi, so this term will be used in the future.

In accordance with the standard, Wi-Fi networks have an architecture consisting of at least one access point and one client. The access point transmits its network identifier (SSID) using special signal packets. Knowing the network SSID, a client can determine whether a connection to this access point is possible. If two access points with identical SSIDs fall within the coverage zone, the receiver can choose between them based on the signal level data. It is also possible to connect two clients on an ad-hoc principle, which theoretically makes it possible to say that the Wi-Fi standard is oriented towards decentralized networks, but this is not quite so.

In fact, the standard does not describe the mechanism for switching the network node from the access point mode to the client mode. That is, if two or more clients are on a network with the same SSID, in most cases they will not be able to communicate in peer-to-peer mode, even if it is more efficient for them in terms of data transfer speed. Such an assertion is easy to verify in any home Wi-Fi network: removing clients from the access point inevitably leads to loss of data transmission speed, even if they are in close proximity to each other. If one wants to increase the data transfer speed, one needs to move both clients closer to the access point.

Nevertheless, the standard does not explicitly prohibit the creation of decentralized mesh networks not from clients, but from access points (in particular, it is described in the IEEE 802.11s standard), which makes it possible to use this technology when designing the architecture of the Industrial Internet of Things. To date, such solutions are implemented at both hardware and software levels.

Analysis of hardware solutions for the creation of cellular networks based on Wi-Fi technology displayed that most of these devices belong to the class Small Office, Home Office (SOHO). Only a small number of manufacturers position their network devices as industrial. Among the most famous industrial solutions, it is possible to identify specialized Cisco devices, for example, Cisco Industrial Wireless 37xx and Cisco Aironet, many devices from Mikrotik, Juniper Networks, etc.

Such devices have proven themselves on the market, but according to the authors, their use to create an Industrial Internet of Things within the proposed concept of modular equipment is not entirely justified. First, such devices will not allow creating a completely decentralized network, where each client can directly communicate with everyone. Secondly, these solutions are quite expensive and often carry excessive functionality that will not be claimed in the area of the projected CPPS. Thirdly, all these solutions are proprietary, which calls into question the possibility of their modernization and improvement through the use of equipment from different manufacturers or open source solutions.

Next issue is the consideration of software implementations of mesh networks based on Wi-Fi technology. It should be noted that most implementations are based on the already mentioned IEEE 802.11s standard, but do not have a binding to any particular hardware vendor.

First of all, it is necessary to mention the project “open80211s” (https://github.com/o11s/open80211s). The goal of this project is to create a full and open implementation of this standard for use on devices running operating systems of the Linux family. Using `open80211s` allows overcoming the limitation on the use of mesh mode on devices of different manufacturers at the firmware level. The project has a good repository structure with source codes, a large number of developers, but, nevertheless, it is practically not being developed or used today. In particular, it should be noted that the last commit to the main branch (master branch) of the repository on `github.com` was made five years ago.

In conclusion, it should be noted that the idea of using open source and hardware solutions based on ESP devices family seems pretty promising. Such devices are based on special ESP microchips (like ESP8266 or ESP32) with embedded Wi-Fi interface and ability to execute programs from external flash memory. Considering it as a basis for a mesh network we find several projects—“painlessMesh” (https://github.com/gmag11/painlessMesh) and “easyMesh” (https://github.com/Coopdis/easyMesh). Both projects provide firmware and API that allows one to build an ad-hoc wireless mesh network. The network implemented with these projects has no IP communication instead of a simple unique chipid identification is used. Moreover, messaging is based on a simple JSON format that seems useful for easy integration with external systems. However, practical usage of these projects appears to be incompatible with real production cases.

First of all, there are performance issues that constrain the number of messages per minute limited by processing power, a memory of the controller and messaging format. Along with that, there is no protection from overloading and self-stabilization. Due to the security issue, a password is used that requires to configure a node before its integration and goes against ad-hoc networking principles. Giving the above, such solutions are more suitable for home automation rather than the industrial field.

B. Cellular Internet of Things

It is obvious that the structure of mobile cellular networks is very similar to the structure of mesh networks. This similarity has led to the emergence of a whole new direction in the development of decentralized networks, called the Cellular Internet of Things. To date, there are at least three technologies related to this class: EC-GSM-IoT, NB-IoT, and LTE-M.

The first of them—EC-GSM-IoT (Extended Coverage–GSM–Internet of Things)—is a standard for low-speed data exchange in 2G, 3G, and 4G mobile networks. EC-GSM-IoT regulates communication of various smart things with the purpose of data transfer between them. The standard allows utilizing all the advantages of GSM-networks available today: device authentication, confidentiality of data transmission, data integration, etc.

In addition to that, this standard allows overcoming the limitations associated with increased power consumption of devices that work as classic mobile GSM devices (phones, smartphones, tablets, etc.). The latter is due to the specifics of smart things, as a rule, they all do not need to maintain a continuous connection and do not require high data rates.

The other two technologies are practically the same as the EC-GSM-IoT. LTE-M is the logical development of LTE networks for IoT while maintaining backward compatibility with the LTE standard. While NB-IoT uses a different type of modulation and is not compatible with LTE networks.

In general, all presented approaches are quite interesting and promising, but not suitable for the implementation of industrial networks. Despite the fact that the developers of standards speak about the possibility of using their technologies in the industry, in fact, there is some shift in the direction of logistics, tracking, diagnostics, supply chain management, etc. And the main areas where these protocols can be applied are the oil and gas industry and food production. The use of such technologies in instrument engineering and machine building is extremely doubtful, according to the authors.

The main disadvantages of mobile networks application in a subject area considered in this paper:

- the low latency of the network, which is permissible for solving of sensors' data receiving problems but not permissible for control (even non-real time one)
- binding to external data networks: failure of the mobile network will lead to the failure of the entire network of the enterprise.
- binding to the equipment of a specific manufacturer.
- the necessity of using physical SIM-cards for access to the network (the Embedded SIM standard has not yet become widespread).

C. Mesh networks based on the IEEE 802.15.1 standard

The IEEE 802.15.1 standard defines the physical layer parameters for various devices joined by a wireless personal network (Wireless Personal Area Network, WPAN). Usually, it is denoted by the term Bluetooth (or Bluetooth Low Energy—BLE). Bluetooth combines the inexpensiveness of equipment, independence from any specific equipment manufacturer, the simplicity of the protocol, sufficient performance, the ability to connect nodes at distances up to 100 m, the presence of an integrated mechanism for ensuring data confidentiality, and sufficiently low delays in transmission in the network.

Since 2017, the Bluetooth standard officially supports mesh network mode, that makes it extremely promising for Industrial Internet of Things (IIoT) systems. There is already a sufficient number of successfully implemented projects for the IIoT, where BLE was chosen as the basis for building a network. All of the above speaks for BLE application in a CPPS core network as it meets all the requirements stated in Section I.

However, this technology still has drawbacks. Firstly, only the latest versions of the Bluetooth 5+ standard implements the mesh mode. Thus, it is still difficult to find ready-made solutions (or even systems on the Crystal, SoC) for the latest version. Secondly, there is no common standard for the transmission of IPv6 traffic via Bluetooth 5+. There is a standard RFC 7668 and a number of researches devoted to the standard description and examples of mesh-networks created on its basis [13] but it has not yet been widely spread.

D. Mesh networks based on the IEEE 802.15.4 standard

As well as the Bluetooth standard IEEE 802.15.4 defines the physical layer and media access control (MAC) for wireless personal networks. The remaining levels of the OSI model are not described. There are several implementations of this standard to specify higher levels. The most common technologies based on the IEEE 802.15.4 standard are ZigBee, ISA100.11a, WirelessHART, MiWi, Z-Wave, and OpenThread.

In general, the differences between networks based on IEEE 802.15.4 and IEEE 802.15.1 are the following:

- The IEEE 802.15.4 protocol is designed for a lower throughput. It is about 1 Mbitps for BLE and no more than 260 kbitps for IEEE 802.15.4
- BLE uses one frequency band within 2400–2483.5 MHz while IEEE 802.15.4 allows network nodes to communicate on others non-licensed frequency bands—868.0–868.6 MHz and 902–928 MHz

- Initially, Bluetooth was oriented to the “star” type topology, point-to-point connections with the ability to organize mesh networks appeared only in the latest version of the standard. In the IEEE 802.15.4 specification, both topologies were originally described.
- For a long time, Bluetooth was focused on consumer electronics, and only recently projects that use this technology in industry and production began to appear. Whereas, The IEEE 802.15.4 specification was originally designed for industrial use.

The most promising implementations of the IEEE 802.15.4 specification to create a cyber-physical system for the integration of modular technological equipment are ZigBee and OpenThread. These technologies are very similar, but have the following slight differences:

- ZigBee is more developed because it appeared earlier. It has larger community, more detailed documentation, and more usage examples.
- OpenThread is a more open technology because it was originally developed to attract as many developers as possible.
- OpenThread has more portability. It supports both system-on-chip (SoC) and network co-processor (NCP) designs.
- OpenThread is focused on the transfer of IPv6 traffic and uses the standard 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks). Therefore, each node can be addressed with its IP address. ZigBee was not originally IP-based technology. To transfer traffic to the IP-network, you either need to use edge router or a new separate protocol ZigBee IP, which, however, has not yet found wide distribution.

In conclusion, it should be noted that it is extremely difficult to choose one of these three technologies: BLE 5+, ZigBee, and OpenThread. The search for information in open sources showed that manufacturers of equipment are of the same opinion—many of the solutions on the market support several protocols at the same time.

Nevertheless, in our opinion, the most interesting and promising technology is OpenThread. Thus, it will be used in the future description of the infrastructure of the developed modular equipment interaction network.

IV. Proposed approach

The main feature of the modular industrial equipment is the ability to quickly re-configure any unit of equipment when changing a technological process. Re-adjustment is carried out due to the change of processing tools and external modules. This is achieved by maximizing the autonomy of each processing head or module. Each of them has its own functioning algorithms, as well as a set of sensors and actuators. All processing heads and modules can communicate with each other via a unified protocol, with a set of commands and data minimized, all commands are strictly regulated. The main module is a three-axis chassis that moves the processing heads in three-dimensional space. Naturally, from the control point of view, such a system can not be completely decentralized—with each equipment unit its virtual dispatcher (“digital twin”) is connected, which is a model of equipment and coordinates modules, heads and chassis. It is the presence of the dispatcher that allows quick adjustment of equipment. All heads and modules are configured in the same way, they all know their capabilities and protocol of interaction, so when they are physically connected they find their dispatcher in the network, register their services, and then they are ready to work right away.

This concept works fine if it is imagined that only one unit of equipment exists, but it is not. CPPS can consist of many hundreds and thousands of devices, and the vast majority of these devices is precisely the technological equipment. In this case, as already mentioned above, each piece of equipment, each processing head, each module, and each sensor is connected to a common decentralized mesh network, because they are parts of the same CPPS.

At this moment the problem is traced: if all modules are located to a decentralized network and have the ability to find their dispatcher, how can they determine that this particular controller is physically connected to this module. A usual way solution: to assign this duty to the operator. However, this will break the “zero configuration” principle, that is, the operator should only think about the technological process, not about the configuration or reconfiguration of the equipment. Here an analogy with conventional universal equipment can be given. A worker who performs operations on a lathe, for example, should not think about the information compatibility of the tool and the machine. He has enough physical compatibility. For example, the cutter is physically placed in the tool post. It is this simplicity of readjustment that we want to achieve in our work, not just for “dumb” universal equipment, but for “smart” equipment integrated into CPPS.

From all of the above, it can be concluded that the organization of a CPPS mesh network, in which modular equipment is used, is not a trivial task. Therefore, the rest of the section will be devoted to the deployment of the OpenThread test network, the description of the general CPPS architecture built on it, as well as principles of the modular equipment included in the CPPS.

A. OpenThread testbed

As already noted in the previous section, OpenThread technology was chosen as the core CPPS network by its set of merits and demerits. This wireless data transfer protocol is a BSD licensed open-source implementation of the Thread network protocol originally developed by Google Nest. A distinctive feature of OpenThread is maximum portability, while according to the specification, various designs can be implemented, both purely hardware and software-hardware. Such flexibility allows one to first create a support network structure using virtualization technologies, test and debug it, and then move it to physical devices.

To deploy the virtual network, a set of debugging tools was used, located in the OpenThread repository (https://github.com/openthread/openthread). In particular, the assembly of a network client for the x86 architecture was used with the emulation of the radio module via a network card with messages transmission using the UDP protocol. The primary tasks were the construction of a simple network and working out, on its example, an algorithm for the nodes interaction. To simplify the task, the original network structure consisted of three independent nodes (Fig. 1). Each node of the network was deployed the same way, which allowed to automate this process in a virtual environment.

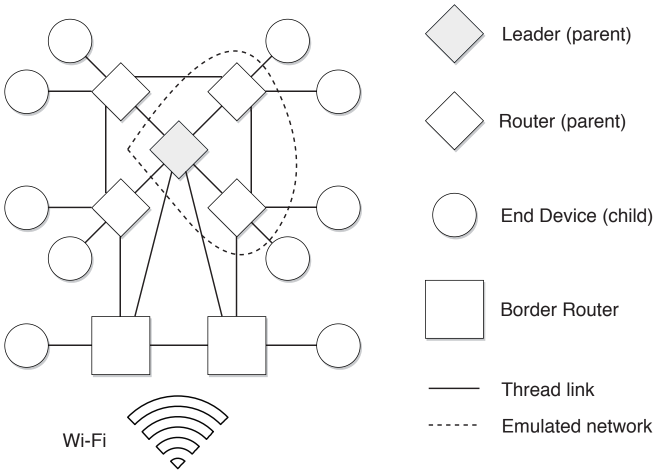


Fig. 1. OpenThread architecture

In accordance with the protocol specification, there are two types of nodes:

- A Router that never turns off the transceiver, participates in packet transmission over the network, and is also a security node for all new connected devices.
- End Device, which mainly interacts only with its router, does not transfer data over the network and can turn off its transceiver to save power.

In addition to this, all devices in OpenThread mesh network can be divided into the Full Thread Device and the Minimal Thread Device. The Full Thread Device never turns off its radio transmitter, subscribes to the all-routers multicast address, and maintains IPv6 address mappings. Full Thread Devices consists of Router, Full End Device (can not self-elect as a router) and Router Eligible End Device. The latter works as an End Device, but it can become a Router if the number of Routers on the network is less than 16, or if a Full End Device appears next to it.

A Minimal Thread Device does not subscribe to multicast traffic and forwards all messages to its Parent. There are two types of Minimal Thread Devices: Minimal End Device—radio transceiver always on, does not need to poll for messages from its parent and Sleepy End Device—normally disabled, wakes on occasion to poll for messages from its parent. Obviously, the role of Minimal Thread Device is mainly used for nodes that are autonomous devices with battery power source. In scenario considered in this paper the using of Minimal Thread Devices is not planned.

Of the entire set of routers, one of them (usually the first turned on) is assigned as Leader. Leader aggregates and distributes information about the configuration of the entire network. There can only be one Leader in one PAN (Personal Area Network). In order to ensure reliability and fault tolerance, the Leader's role dynamically moves from one node to another, that is, any router can self-elect as a Leader. Also, one Border Router can be assigned to the network, which is responsible for the interaction and transfer of IPv6 traffic between the mesh network of OpenThread and other networks (Wi-Fi, for example).

Thus, the test network will consist of one Router, with the assigned role of Leader and two Router Eligible End Devices, which will demonstrate the ability of changing the role when disconnecting/connecting nodes.

As a test environment for virtualization, a server with the following technical specifications is used: 2 x Intel Xeon E5620 CPU @ 2.40 GHz, 32 GiBRAM, Intel 82575EB Gigabit Network. Operating system: Ubuntu 16.04.2 LTS, GNU/Linux 4.4.0-64 x86-64. As a hypervisor, Oracle VirtualBox 5.2.16 and the Vagrant 2.1.2 virtual environments management system were used.

The virtual machine configuration file is downloaded from the official github repository using the command

```
$ git clone https://github.com/openthread/openthread
```

After this, deploying of the guest operating system and configuring of OpenThread client are executed:

```
$ cd ˜/openthread/etc/vagrant
$ vagrant up
```

During the installation process, the configuration of the basic guest machine running under the operating system Ubuntu 14.04 LTS is performed. This machine is a template by which an arbitrary number of OpenThread nodes can be created. To connect to a virtual machine, one should use the following command:

```
$ vagrant ssh
```

Then creating the first node using the console client of the network can be started. It should be noted that in this example all actions for node configuration and interaction over the network will be done manually by entering commands into the shell client. In the future, all these actions will be automated due to the fact that each shell command is just a call to a function C Thread API. The console client launch:

```
$ cd ˜/src/openthread
$ ./output/x86_64-unknown-linux-gnu/bin/ot-cli-ftd 1
```

The first parameter indicates that the transmission will use radio emulation via the UDP protocol. Specifically, this parameter point on the port number to be used for communication ($9000 + 1 = 9001$ port). For subsequent nodes this number must be different. Naturally, in a real network, this configuration is not required. All devices are configured in advance in a uniform manner and operate on one of the predefined radio channels IEEE 802.15.4. Non-reserved values are in the range of 11–26. First nodes are configured with these commands:

```
> panid 0x1234 # to set the unique network id
> ifconfig up  # to start network IPv6 interface
> thread start # to start Thread protocol
```

After Thread protocol was started, the first node checks the network for the presence of other nodes and, since there are no other nodes in the network, assigns the role of Router and Leader. The second and third nodes are configured using the same commands. The only difference is that during the initial scan of the network, the nodes see that there is already one Router on the network, therefore they initially assign the role of Child, which automatically changes to the Router after a two-minute timeout. This is due to the fact that the Thread network tries to maintain the number of Routers in the network in the range 16–23, respectively, until the minimum value is reached, each newly connected node will change its role on the Router.

The demonstrated example of the configuration of the OpenThread test network clearly shows that it is possible to perform a complete preliminary configuration of devices without the need to manually connect the new device to the network. At the same time, it can be noted that the configuration is a very simple procedure, the connection to the network is very fast, and after the launch, any network node immediately identifies all the neighboring nodes and can interact with them via IPv6 protocol.

B. Communication protocol

In the previous section, the procedure for physically deploying the decentralized backbone CPPS network was considered. The result is a ready-made backbone network in which nodes can exchange raw data. However, it should not be forgotten that this system is not an ordinary sensor network, where only the simplest protocols of the application layer are used, for example, the MQTT message queue.

Of course, the presence of all possible sensors monitoring the production process is implied, but the basis of CPPS is a modular industrial equipment. On the one hand, it consists of autonomous and independent modules located in a common mesh network. On the other hand, all these modules are able to organize stable hierarchical formations to perform specific tasks of production. From the last statement it follows that for the effective operation of such a heterogeneous network, a more complex protocol for two-way interaction is needed.

This protocol should to be considered in more detail. Consolidation of modules is carried out around the base module, which acts as a dispatcher. Since it is assumed that to ensure the flexibility of equipment, it must be maximally unified, each piece of equipment includes in its composition a mandatory module—a 3-axis chassis that moves the processing heads in workspace. Of course, many types of processing require the possibility of moving the working element to 4 or 5 coordinates, but, as practice shows, in many even professional 5-axis machining centers, the 3-axis chassis is used as the basis, and additional coordinates are executed as a separate module. Accordingly, from a control point of view, it is the 3-axis chassis that acts as the dispatcher around which one or another unit of equipment is configured.

Each dispatcher has a registry that is part of a distributed JSON-based repository. The registry consists of slots and only one module can be registered in each of them. In its turn, the totality of all registers is a "digital twin" of CPPS and is stored in the cloud. For uniformity in the same scheme, a digital model of devices that are not industrial equipment is created (different sensors of the production process, at first). The sensor is a dispatcher with one slot without the possibility of re-registration (Fig. 2).

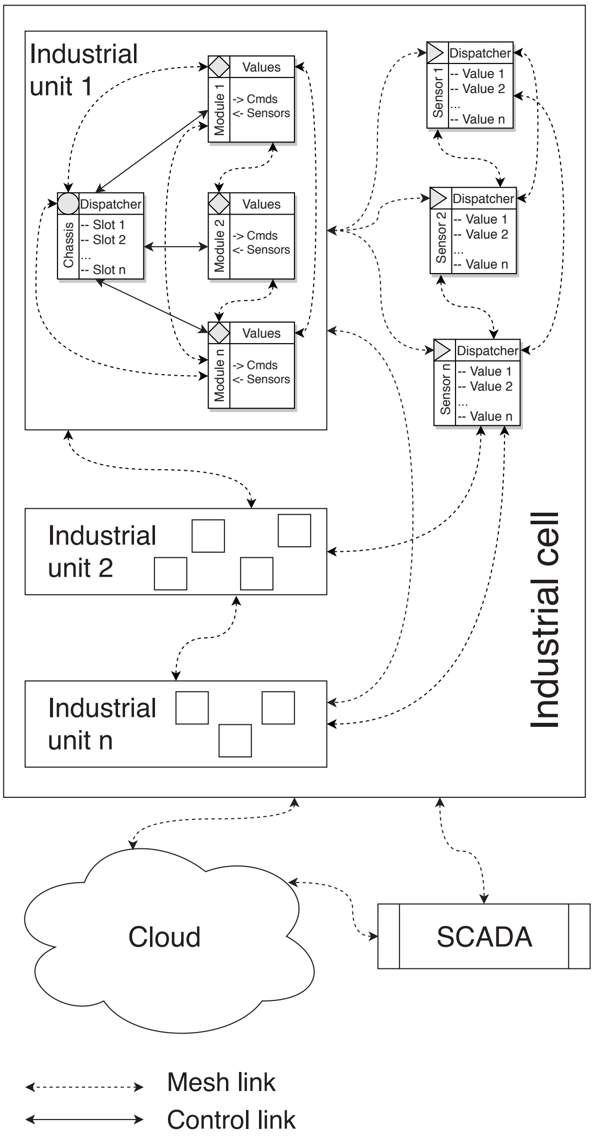
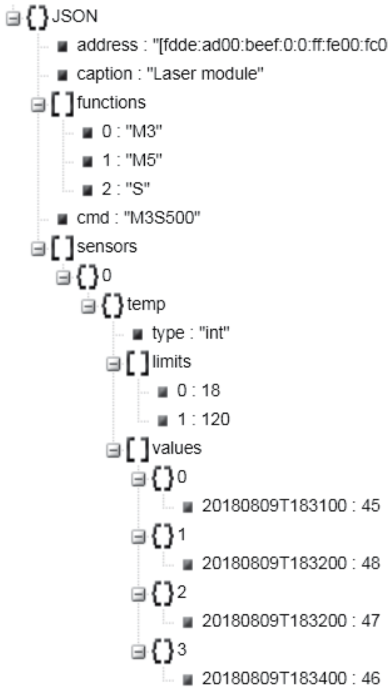


Fig. 2. General CPPS architecture

Slot is a "key-value" type record from which dispatcher retrieves the required data. Slot consists of from:

- an address;
- a module/sensor name;
- functions;
- a return value;
- a range of the return value.

The address is the IPv6 address and port that are required to communicate with the module. The name must be in string or integer format. Available functions are described by a set



```
{}JSON
  address : "[fdde:ad00:beef:0:0:ff:fe00:fc00]:9001"
  caption : "Laser module"
  []functions
    0 : "M3"
    1 : "M5"
    2 : "S"
  cmd : "M3S500"
  []sensors
    {}0
      {}temp
        type : "int"
        []limits
          0 : 18
          1 : 120
        []values
          {}0
            20180809T183100 : 45
          {}1
            20180809T183200 : 48
          {}2
            20180809T183200 : 47
          {}3
            20180809T183400 : 46
```

Fig. 3. Example of a slot

of G-codes and M-functions in accordance with ISO 6983-1 and ISO/TR 6983-2. For example, the milling head will be able to work with the commands M3 and M4 (starting the spindle clockwise and counterclockwise), M5 (spindle stop) and S (rotation speed); for the laser head—this will be the commands M3 (turn on the laser), M5 (turn off) and S (set the power in percents); for the tool store—M6 (change tool) and T (choose the tool from the store by number). In this case, all the commands related to moving the head in three basic coordinates are determined by the chassis module.

There can be more that one return value. The description of each of them includes a type, a range, and an array of values with timestamps. The dispatcher controls the output of each value beyond the permissible limits, and in the event of such a situation, analyzes the error and makes a decision whether or not the equipment can continue to operate. An example of a slot is shown in Fig. 3.

The procedure for registering the module in the dispatcher's registry needs to be explained. From the protocol point of view, there are no problems here: simply transferring the binary JSON message through the message queue and then writing it to the distributed store [14]. However, it should be noted that all dispatchers and all modules are in a single self-organizing mesh-network. The question arises—how the module will determine to which specific dispatcher it is connected physically.

The following is proposed. Each module is physically connected using 3-wire interface, and two wires are used as power supply and ground lines while third wire is used as a safety line. The safety line connects all equipment units modules as wired and logic circuit of an AND type (Fig. 4). The safety line utilizes the pull-up resistor. Due to huge resistance between power supply line and ground and relatively low resistance, that equals connected resistor between safety line and power supply, the voltage on safety line equals to power supply voltage. Thus the low and high logic levels are formed. As it was spoken before, all modules are connected to safety line and can pull it to ground. Consequently, the high logic level on the safety line will be presented only when all the modules switch the outputs to high logic level. If any of the modules ground the safety line, the low logic level on the safety line will be presented and none of modules will be able to switch it back to high.

Thus, the main task of the safety line is to register emergency situations. In case of an accident in any of the modules, it simply connects the safety line to the ground, which is a signal for other modules to stop working and get into recovery mode after a failure. It should be noted that the emergency stop button is also connected to the same line, which is mandatory for any industrial equipment, as well as all safety doors, if they are provided by the design. However, during the initialization of modules, the safety line can be used to register new modules. The newly connected module firstly connects to the OpenThread network, then sets the safety line to a low logical level. The dispatcher detects this, then gets a list of all neighbors closest to it, chooses those that have the status "not connected" among them, then asks the first of them to set the line back to a high logical level. If the level has changed, then the dispatcher and the module are connected to the same line, therefore, the module can be registered in the registry. Otherwise, the dispatcher moves to the next module in the list. At the same time, it is known that all dispatchers can communicate with each other, so one needs to establish a rule that regulates the order of initialization of dispatchers, i. e., if one of them spends filling the slots of its registry, the others should be in the standby mode and do not accept connections from the modules. It is also clear that the process of connecting a new module can be carried out only in the case, when the equipment is in standby mode.

V. DISCUSSION

Currently, the proposed approach is intended for small-batch production companies. It is really convenient for such companies to have a set of universal and easy to reconfigure equipment that can be joined into various chains according to a technological process. The suggested idea allows one to solve shop floor organization and equipment placement issues, as each equipment unit based on the universal chassis only depend on the electrical network. Moreover, the ability to remove unused modules optimizes equipment storage space and gives the opportunity to move quickly in case of company relocation. This concept can be called "a production office" due to the ease and speed of production deployment at a new placement like simple office equipment. Upon that, there is practically no equipment on the market that would have such properties

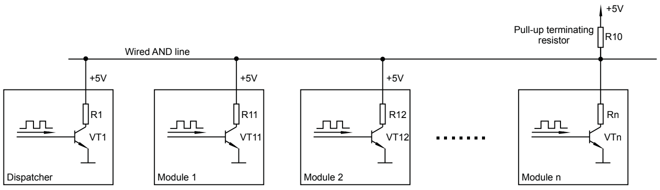


Fig. 4. Wired AND line circuit

nowadays. The only exception is industrial 3D printers, which can be put into operation in a matter of hours. They do not have special requirements for deployment though. Moreover, 3D printers are connected to the wireless network and can work on ordinary office premises.

On the other hand, the idea to extend this approach for large production companies makes sense. Indeed, features of wireless self-organizing mesh-networks can be useful in mass production. Mesh networks are already partially applied in the form of sensor networks covering large production areas, where laying additional cable lines is unprofitable. The benefits, in this case, are 50–90 % compared to the cost of a wired network [15]. However, we believe that having a common network for all industrial devices is just as important as collecting data from sensors. There are three problems that have to be solved when implementing a common mesh network that joins not only sensors but also industrial equipment. First, the IEEE 802.15.4 standard does not imply the possibility of data transmission over sufficiently long distances without significant loss of speed. It is especially important for automated (automatic) guided vehicles or mobile robots used in production. To solve this issue stationary retransmitters located at the node points of workshops and other production facilities can be applied. Secondly, the security issue is still not solved. IEEE 802.15.4 protocol describes a physical layer of the OSI model, but it is not enough to prevent complex attacks. Currently, we are carrying on the research on the development of a two-level protocol. In this protocol, a message queue is applied to transfer data from autonomous sensors to a common network and for equipment modules communication. At the CPPS level blockchain technology is employed as well as smart-contracts [16]. Thirdly, the issue of radio frequency range choice still exists. The majority of modern wireless networks uses a frequency of 2.4 GHz. Obviously, this frequency is very congested and noisy. Globally, this problem can only be solved by allocating exclusive frequency ranges for industrial automation. However, for small production, it is possible to use local screening of production facilities.

VI. Conclusion

The main issue in organizing the work of CPPS is a distributed network of modular equipment that can include thousands of devices, sensors and controllers, which must communicate with each other and react to events in a timely manner. The composition complication of the production equipment leads to the connection complication between the equipment units. At some point, the standard "star" topology is obviously inefficient for such networks, since there is a risk of overloading the central node. In addition, this topology directly contradicts the concept of decentralized management. This paper briefly presented the network architecture based on the application of mesh-networks, in particular, on the OpenThread technology. The possible approaches to the implementation of mesh networks for industrial use have been described. Further, the mesh-network deployment architecture is proposed and its virtual model is tested. The description of interaction of network nodes and the procedure for registering a new module in the register of the dispatcher are given.

Obviously, this paper shows the first stages of development and implementation of the proposed architecture. Of course, there are a number of architectural limitations related both to the technical features of the technologies used and to the security of the described interaction, which will need to be solved consistently in further research.

VII. Acknowledgements

This work was carried out under project no. 617026 "Technologies of cyber-physical systems: management, computing, security".

References

[1] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of microservices architecture pattern to create a modular computer numerical control system," in *2017 20th Conference of Open Innovations Association (FRUCT)*, April 2017, pp. 10–19.

[2] ——, "Modular industrial equipment in cyber-physical production system: Architecture and integration," in *2017 21st Conference of Open Innovations Association (FRUCT)*, Nov 2017, pp. 1–9.

[3] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, Nov 2014, pp. 634–638.

[4] I. M. J. Haxhibeqiri, E. Jarchlo and J. Hoebeke, "Flexible Wi-Fi communication among mobile robots in indoor industrial environments," in *Mobile Information Systems*, 2018, p. 19.

[5] F. Kauer, E. Kallias, and V. Turau, "A dual-radio approach for reliable emergency signaling in critical infrastructure assets with large wireless networks," *International Journal of Critical Infrastructure Protection*, vol. 21, pp. 33–46, 2018.

[6] T. Dang and C. Devic, "OCARI: Optimization of communication for ad hoc reliable industrial networks," in *2008 6th IEEE International Conference on Industrial Informatics*, July 2008, pp. 688–693.

[7] P. D. Wegener, *GERMAN STANDARDIZATION ROADMAP Industrie 4.0 Version 3*. DIN e. V., 2018.

[8] K. Ahmed, N. S. Nafi, J. O. Blech, M. A. Gregory, and H. Schmidt, "Software defined industry automation networks," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2017, pp. 1–3.

[9] K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN-enabled tactical ad hoc networks: Extending programmable control to the edge," *CoRR*, vol. abs/1801.02909, 2018. [Online]. Available: http://arxiv.org/abs/1801.02909

[10] U. Graf, R. Heidel, and D. G. Kadel, *DISCUSSION PAPER. Network-based communication for Industrie 4.0*. BMWi, 2016.

[11] L. Bösch. Spokeswoman Audi IT. Audi and Ericsson to pioneer 5G for automotive manufacturing. [Online]. Available: https://www.ericsson.com/en/press-releases/2018/8/audi-and-ericsson-to-pioneer-5g-for-automotive-manufacturing

[12] A. Neumann, L. Wisniewski, R. S. Ganesan, P. Rost, and J. Jasperneite, "Towards integration of industrial ethernet with 5G mobile networks," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, June 2018, pp. 1–4.

[13] P. Narendra, S. Duquennoy, and T. Voigt, "Ble and ieee 802.15.4 in the iot: Evaluation and interoperability considerations," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 170, pp. 427–438, 2016, cited By 4.

[14] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "Performance evaluation of the message queue protocols to transfer binary json in a distributed cnc system," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, July 2017, pp. 357–362.

[15] J. Colpo and D. Mols, "No strings attached," *Hydrocarbon Engineering*, vol. 16, pp. 47–52, 2011.

[16] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of blockchain and smart contracts for machine-to-machine communications in cyber-physical production systems," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, May 2018, pp. 13–19.