Industrial networks and IIoT: Now and future trends

Industrial Networks and IIoT: Now and Future Trends

Chapter · July 2020

DOI: 10.1007/978-3-030-42500-5_1

3 authors:

Alparslan Sari
University of Delaware
4 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Alexios Lekidis
Intracom Telecom S.A.
24 PUBLICATIONS **58** CITATIONS
SEE PROFILE

Ismail Butun
Chalmers University of Technology
49 PUBLICATIONS **1,039** CITATIONS
SEE PROFILE

Industrial networks and IIoT: Now and future trends

Alparslan Sari, Alexios Lekidis, and Ismail Butun

Abstract Connectivity is the one word summary for Industry 4.0 revolution. The importance of Internet of Things (IoT) and Industrial IoT (IIoT) have been increased dramatically with the rise of industrialization and industry 4.0. As new opportunities bring their own challenges, with the massive interconnected devices of the IIoT, cyber security of those networks and privacy of their users have become an important aspect. Specifically, intrusion detection for industrial networks (IIoT) has great importance. For instance, it is a key factor in improving the safe operation of the smart grid systems yet protecting the privacy of the consumers at the same time. In the same manner, data streaming is a valid option when the analysis is to be pushed from the cloud to the fog for industrial networks to provide agile response, since it brings the advantage of fast action on intrusion detection and also can buy time for intrusion mitigation.

In order to dive deep in industrial networks, basic ground needs to be settled. Hence, this chapter serves in this manner, by presenting basic and emerging technologies along with ideas and discussions: First, an introduction of semiconductor evolution is provided along with the up-to-date hi-tech wired/wireless communication solutions for industrial networks. This is followed by a thorough representation of future trends in industrial environments. More importantly, enabling technologies for industrial networks is also presented. Finally, the chapter is concluded with a summary of the presentations along with future projections of IIoT networks.

Alparslan Sari (Ph.D. Candidate)
University of Delaware, Department of Electrical and Computer Engineering (ECE),
Cybersecurity Research Group, Newark, DE, USA
e-mail: asari@udel.edu
Alexios Lekidis (Ph.D.)
Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: alekidis@csd.auth.gr
Ismail Butun (Ph.D.)
Chalmers University of Technology, Department of Computer Science and Engineering (CSE),
Network and Systems Division, 412 96 Gothenburg, Sweden
e-mail: ismail.butun@chalmers.se

1 Introduction

Industry 4.0 revolution can be summarized with one word: 'Connectivity'. Connectivity will enable intelligent production with the proliferation of IIoT, cloud and big data. Smart devices can collect various data about indoor location, outdoor position, status information, usage patterns of the clients, etc. They have the ability not only in gathering information, but also sharing the information amongst intended peers. This will be beneficial in building an efficient manufacturing process in industrial environments and also in helping with the planned preventative maintenance on machinery. The other benefit is in identifying errors in the production pipeline as quickly as possible since it is an important factor to reduce the production and maintenance costs. Industry 4.0 is also focusing on optimization problems in the industry by using smart devices to utilize data-driven services. Industry 4.0 and IIoT are used for complex task sharing, decision making based on collected data, and remote access to machinery. Massive connectivity of the things and data collection/sharing capability of those promotes security to be a major requirement for the IIoT and Industry 4.0 concepts.

Semiconductor transistors are introduced in the late 1940s and led the microprocessor revolution in the 1970s. Intel produced MOS (Metal-Oxide Semiconductor) based 4 bits 4004 microprocessor in 1971. It had 2,250 transistors, 10,000nm MOS process and area of 12mm^2. Figure 1 [1] shows the plot of MOS transistor counts by years. In the plots, the exponential increase of transistor counts validates Moore's law - transistors in an integrated circuit doubles every two years. Currently, transistor counts jumped to $9-40$ billion with $7-12nm$ MOS process and area of $100-1000mm^2$. Figure 2 and 3 shows the evolution of the flash memory and RAM with transistor count vs. date plot. Groundbreaking technological advances in electronics started the information age which triggered major changes in communication technologies such as the Internet (connectivity), advanced machinery, and software development, etc. Internet became the global communication hub, enabling us to exchange information instantly. Advanced electronics produced smart devices with a smaller size, which constitute today's 'things' of IoT and IIoT.

IoT is the proliferation of smart devices such as tablets, phones, home appliances such as TVs, and other sensors, etc. The benefit of using smart devices at home would be reducing electric bills and time savings etc. Managing resource usage based on sensors or scheduling heavy-duty tasks like running dishwasher, washing machine or dryer when the electric consumption is the cheapest. IoT devices are commonly used by the hobbyist or another consumer usage, and even in industry. However, IIoT is designed for heavy-duty tasks such as manufacturing, monitoring, etc. So, IIoT uses more precise and durable (heat/cold resistant) devices, actuators, sensors, etc. Both IoT and IIoT have the same core principles such as data management, network, security, cloud, etc. The main differences between IoT and IIoT are scalability and the volume of generated data and how data has been handled. Since IIoT devices generate massive amount of data, IIoT requires data streaming, big data, machine

[1] Figures 1, 2, and 3 are illustrated based on data from:
https://www.wikiwand.com/en/ Transistor_count

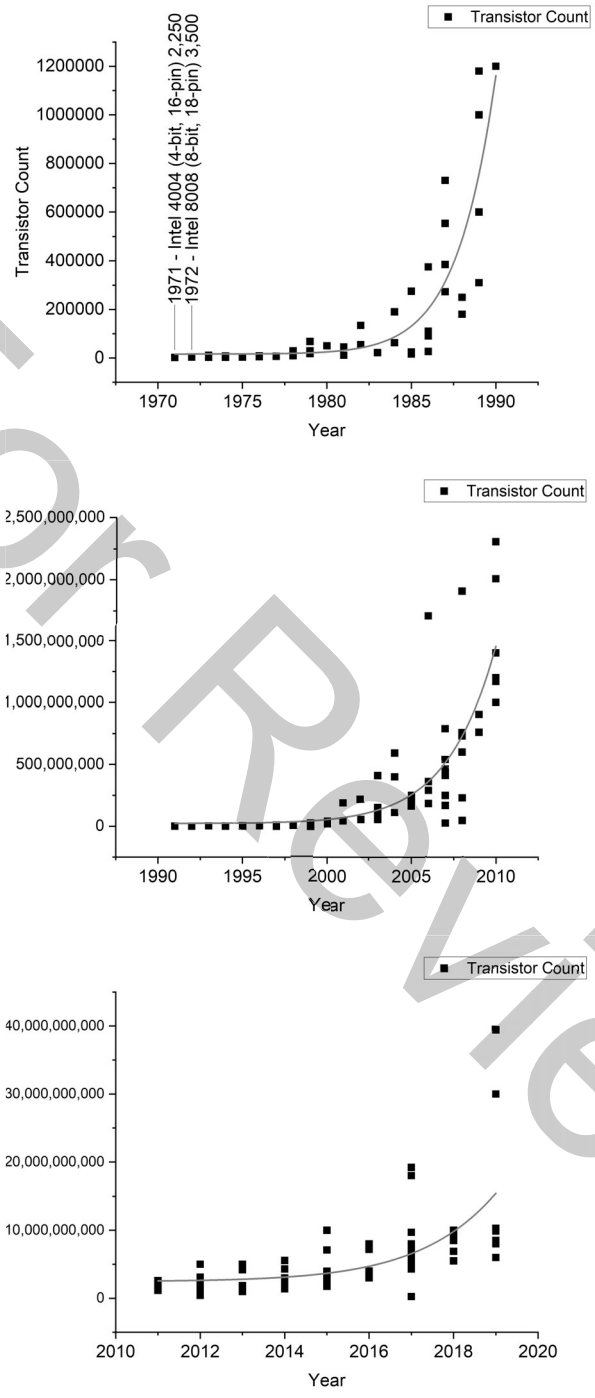

Fig. 1: Figure shows transistor counts in microprocessors by date between 1971-1990 (top), 1991-2010 (middle), and 2010-2019 (bottom).

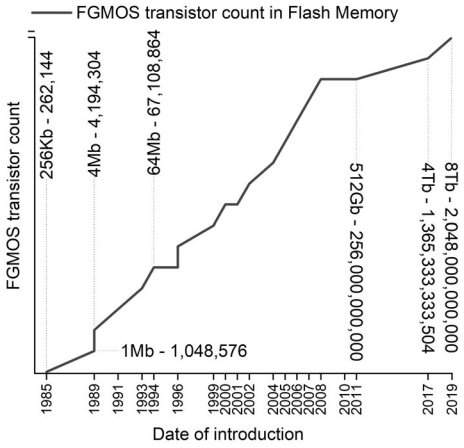

Fig. 2: FGMOS transistor count in Flash memory over years with capacity increase.

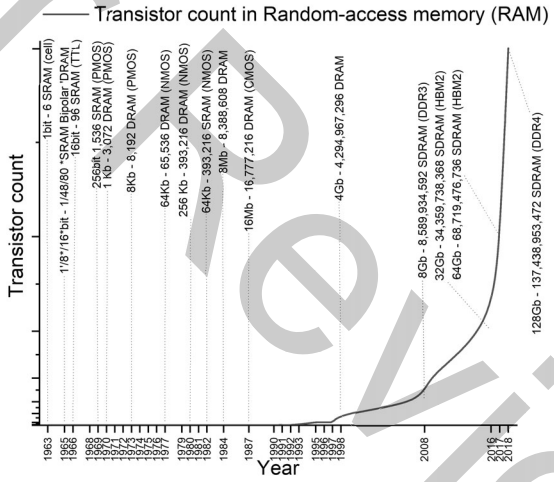

Fig. 3: Transistor count in RAM over years with capacity increase.

learning or artificial intelligence practices. In a home network, loss of the generated data would be trivial but in IIoT it is vital. The data in IIoT should be more precise, continuous and sensitive. For instance, considering a monitoring system in a nuclear power plant or a manufacturing facility should be precise, continuous and sensitive to prevent hazardous events. The implementation of IIoT in production lines or other industrial projects, companies are aiming to reduce production or maintenance costs and improve efficiency, stability, safety, etc.

According to a key note speech[2] delivered by Tom Bradicich[3], the 7 principles of the IIoT are provided as follows:

- *Big amount of analog data:* Many sensors generate analog data and this data needs to be digitized to be further treated, analyzed and stored.
- *Perpetual connectivity:* Devices of the IIoT are always connected. There are three key benefits of this: 1) Real-time monitoring is possible. 2) Continuous monitoring can help us to push software-firmware updates and fixes. 3) The benefit of connected devices motivates individuals and organizations to purchase products.
- *Real-time data streaming:* In the industry there are many safety mechanisms are in use, and they are constantly generating data. Considering a nuclear power plant, safety is of utmost critical from the operations point of view. Monitoring requires real-time data streaming since a possible delay in data would cause disastrous events. Hence, real-time data streaming and its aggregation are really important.
- *Data insights:* Data insights (Spectrum of Value) in IoT seeks an answer to the following question: "What are you trying to achieve?"
- *Time-to vs. depth-of -insight trade-off:* It is equivalent to the immediacy-of-knowledge compared to the depth-of-knowledge. For instance, while monitoring or analyzing a nuclear power plant data, immediate attention is required, whereas an analysis of a scientific experiment data (data by CERN or NASA) can take years to uncover scientific problems.
- *Visibility from Big Data:* Once the data is collected and stored in big data environment, later on it should be available whenever it is needed for analysis or other tasks.
- *Edge computing:* Data center class computing and analytics will be shifted to edge (latency, bandwidth, cost, security, duplication, reliability, corruption, compliance, and data sovereignty).

Improvement in big data and data streaming technologies enabled organizations to use Artificial Intelligence (AI) and Machine Learning (ML) products more widely in Industry 4.0. AI and ML applications are emerging in health, education, defense, security, industry, etc. Many technology giants are pouring billions of dollars to develop AI products such as autonomous cars (self-driving cars). Google, NVIDIA, and others are developing computer vision-based self-driving algorithms along with pedestrian detection, collision detection, etc. After the famous Urban Challenge organized by DARPA, autonomous cars brought to the spotlight once again. Automobile companies like Ford, General Motors, Nissan, Tesla, Mercedes, etc. invested billions in R&D. Hundreds of other small companies are building Radars, cameras, computing and communication systems, other sensors, etc. According to the NVIDIA developers' blog, autonomous cars are responsible for the generation of an enormous amount of data involved with the following equipment: Radar, Sonar, GPS, Lidar and cameras. A single forward-facing Radar (2,800 MBits/s) generates approximately 1.26 TB of data per hour. A two-megapixel camera (24 bits per pixel) operating at

[2] Avaliable at: https://www.youtube.com/watch?v=u3IaXvjDiOE
[3] Tom Bradicich, Ph.D., VP and GM, Servers and IoT Systems, Hewlett Packard Enterprise.

30 frames per second generates 1,440 Mbits of data per second (approximately 1TB data per hour [31].

In computer science, the following AI disciplines are utilized for various purposes: Natural Language Processing (NLP), multi-agent systems (coordination and collaboration - distributed resolution of problems, decision and reasoning, learning, planning, simulation), human interactions (learning, chat-bots, expert systems), computer vision, robotics, neuroscience and cognitive science (comprehension and simulation of the brain and nervous system), decision support (game theory, uncertainty, explicability). It is assumed that using AI and ML algorithms will contribute efficiency, cost reduction and effective management of industrial networks. Therefore, following AI-based algorithms are also utilized frequently: heuristics, logical programming, deduction and proof, reasoning, planning, scheduling, and search.

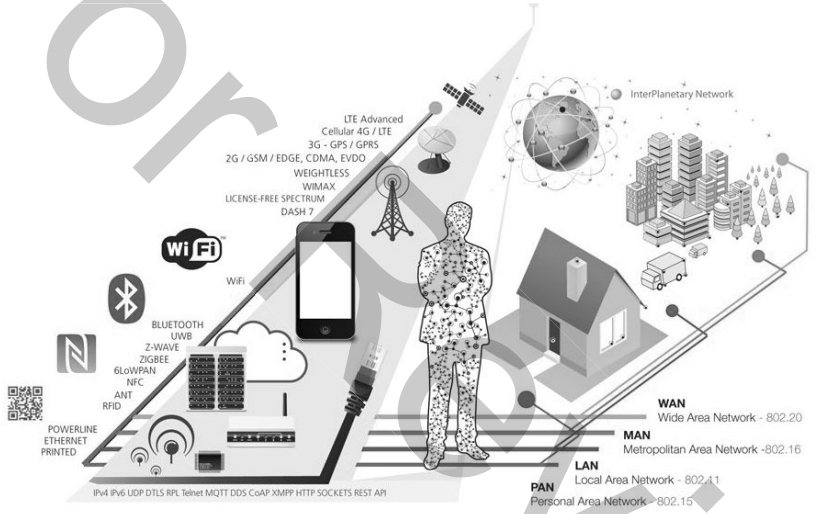

Fig. 4: IoT connectivity diagram[4].

The wired/wireless communication technologies that are being used for IIoT networks are as shown in Fig.4[4] and Fig. 5. They are also summarized here as follows:

BLE: Bluetooth Low Energy (BLE) is a wireless network technology of Personal Area Networking (PAN), which is widely used in smart devices such as phones, watches, wearable electronics, etc. BLE applications can be found in smart home, health and sport industry. Many operating systems have native support for this technology such as Android, BlackBerry, IOS, Linux, macOS, Windows, etc. BLE uses 2.4 GHz radio frequencies.

[4] Source: Postscapes and Harbor Research, available at:
http://postscapes.com/what-exactly-is-the-internet-of-things-infographic/

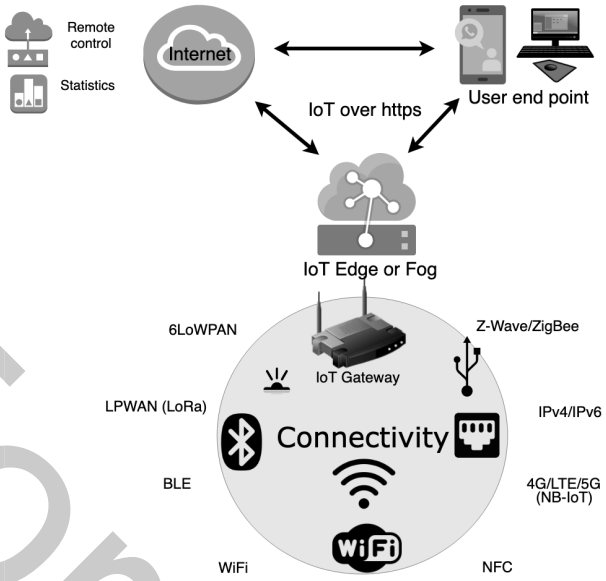

Fig. 5: A subset of IoT and IIoT communication technologies and protocols are illustrated.

WiFi: Wireless Fidelity (WiFi) is an IEEE 802.11 based wireless network technology that is widely used with computers, tablets, smart devices, and vehicles such as cars, buses, drones, etc.

IPv4/IPv6: Internet Protocol versions 4/6 (IPv4/IPv6) can be considered as the backbone of the Internet and other packet-switched based networks. It is one of the most widely used technology in real life.

NFC: Near Field Communication (NFC) is a communication technology which devices can do connection approximately within 4 cm distance. NFC devices are commonly used in electronic payments, key-cards, electronic tickets, etc. Many smartphones are supporting this technology such as Android, Blackberry, IOS, etc.

Z-Wave/ZigBee: Z-Wave is a low energy wireless protocol. Zigbee is also a low power (low energy consumption) wireless communication protocol based on IEEE 802.15.4 standard to create PAN. Z-Wave/ZigBee are commonly used in home automation, medical and industrial applications.

4G/LTE/5G: 4G (fourth generation), LTE (Long Term Evolution), and 5G (fifth generation) are dominating broadband cellular network technologies in nowadays mobile telecommunications.

NB-IoT: Recently, NB-IoT (Narrow Band Internet of Things) rapidly deployed by mobile operators with the proliferation of IoT. NB-IoT is an LPWAN standard that focuses on specifically low cost, energy-efficient, and indoor coverage.

According to one of the leading sector representative[5] of industrial investment companies, IIoT will constitute one of the main pillars of Industry 4.0 and would be described as follows: "The IIoT is a network of physical objects, systems platforms and applications that contain embedded technology to communicate and share intelligence with each other, the external environment and with people." By following this definition and the technological trends, economic predictors[6] deduct this conclusion: "The IIoT has the capacity to significantly boost the productivity and competitiveness of industrial economies, but poor supporting conditions - especially the lack of digital literacy - will hold many countries back".

There will be plenty of application areas for IIoT ranging from smart cities to precision agriculture, smart traffic to smart grid (the future of electric grid), and so on, as shown in Fig. 6.

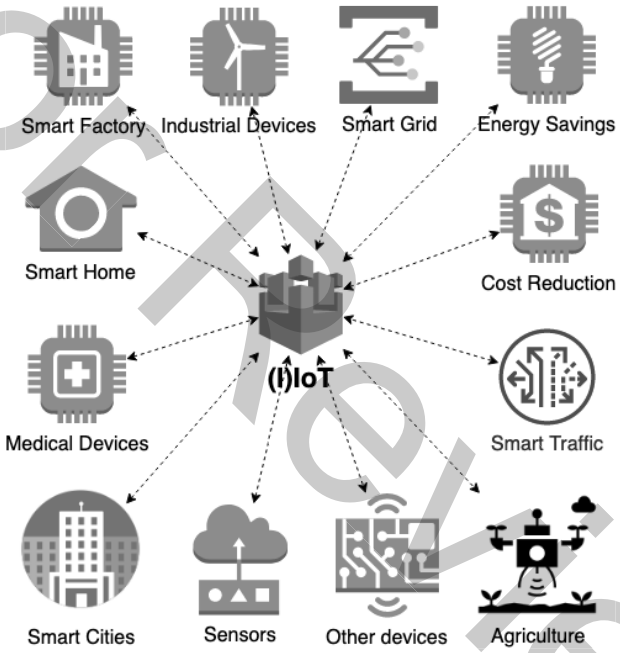

Fig. 6: A subset of IoT and IIoT use cases are illustrated.

As discussed thoroughly in [12], IoT networks are vulnerable to various types of attacks and weaknesses. Due to the absence of a physical line-of-defense, security of such networks is of a big concern to the scientific community. Besides, as cyber-attacks on industrial environments happen more frequently and their results are elevated to a devastating scale, effective intrusion detection is of the utmost

[5] More information is available at: https://www.accenture.com/us-en/
[6] More information is available at: http://www.businesswire.com/news/home/20150120005160/en/Industrial-Internet-Boost-Economic-Growth-Greater-Government

importance of industrial networks. Hence, research regarding Intrusion Detection Systems (IDSs) for industrial networks is arising.

Prevention and early detection of intrusions are important, especially for mission-critical industrial networks, such as smart grid systems.

For instance, data streaming concept provided in later chapters (Chapter-6), might indeed enhance detection/response time in IDSs of the industrial networks. Hence, future IIoT implementations can benefit from data streaming not only to reduce up-streaming data size, but also to improve the security of the network especially by enhancing the overall IDS response time.

Nowadays, there exist example applications in industrial networks (such as smart grid systems), where sensors can generate data at rates of $\sim 3GB$ (Gigabyte)'s per hour [47, 55]. This is way much beyond the Ericsson's and Bloomberg's expectation [2], which was $\sim 280PB$ (Petabyte) of data to be generated by 680 million smart sensors per year (which is equivalent to 1 byte of information per sensor per year). However, current client/server architecture of the traditional networks is not capable of transferring, storing, and processing such big volume of data. Hence, a sampled version of the generated data is used. Secondary benefit of data streaming would be utilization of the unused portioned of the mentioned big data from which IDSs would also benefit.

> Centralized system architectures that gather all the data in the cloud to process it cannot scale, both due to communication bandwidth and issues of scalability of processing accumulated data; e.g. matching patterns of data streams, on-the-fly, is a lower-latency process compared to searching for patterns in terabytes/petabytes of stored data. Due to this, only small fractions of generated data is actually being processed and used in such infrastructures. With this reasoning, it is possible to see that besides helping resolving bottleneck situations, stream processing can enable usage of more data for applications in general and hence for IDSs as well.

So far, a brief overview of industrial networks and IIoT is presented in Section 1. The rest of the chapter is structured as follows: Challenges faced by the practitioners and the researchers while working on industrial networks are presented in Section 2. Section 3 projects the future of industrial networks, including the technologies that can be adopted by. Whereas, Section 4 presents enabling technologies for industrial networks. Finally, Section 5 concludes the chapter.

2 Challenges in industrial networks

Industrial networks are subject to many technical challenges that need to be considered before, during, or after any implementation. These challenges are as follows but not limited to:

2.1 Wireless coexistence

Wireless coexistence stands for the safe operation of the devices that are using wireless technology for the purpose of their communication. Especially, 'signal interference' is the main problem against coexistence of different wireless communication standards, which might cause fading of the signals if they are operating in the same frequency bands. This is on top of the existing wireless communication problems such as reflection, refraction, diffraction, scattering and *free space path loss* of the signals on air[7]. Eventually, wireless signal interference might cause packet drop, data loss, jitter and delay in transmission, and an asynchronization between the two communicating parties. Therefore, while designing IIoT, wireless coexistence of the devices should be maintained, for instance by providing enough physical distance in between the devices, and or by efficiently dividing ans sharing the frequency spectrum that is being used.

2.2 Latency

Latency, sometimes referred to as delay, (for the industrial networks) is the time passed between the release time of a specific command and the start time of the execution for that specific command. In some specific cases, it might be also referred to as the time passed in between the data collection and the output of reaction. For instance, even with the existence of cloud-end implementation, Ferrari *et al.* have shown that, by using inexpensive industrial grade IIoT devices, the round-trip latency of IIoT applications can be less than 300*ms* for inter-continental communications and less than 50*ms* for intra-continental communications [30]. Although, any figure below 300*ms* is considered as sufficient for telephony communication to avoid undesired "talk-over" in conversations, every *ms* counts when the industrial networks are considered. Especially real-time response might be needed for the installments that are involving high speed machinery with safety requirements.

These important aspects *(wireless coexistence* and *latency*) of wireless communication, which is the enabling technology for IIoT and IoT networks, are thoroughly discussed in Chapter-2 of this book. Especially, next generation wireless technologies, such as 5G and the ultra-reliable low-latency communication (URLCC) are presented including the theoretical limits and the *latency*. The trade-offs in low-latency communication for receivers with computational complexity constraints are carefully discussed. Accordingly, it is stressed that, for mission critical IIoT deployments such as the factory implementations requiring real-time response, the latency is utmost important. With very stringent latency requirements and low-complex IIoT receivers, the time required for the decoding of a packet (should not constitute a computationally demanding operation) must be also considered while analyzing the total latency.

2.3 Interoperability

Interoperability refers to the essential capability of various computerized merchandises or systems to promptly connect and exchange data with each other, without facing any restriction. There are two types of interoperability issues defined for IIoT [33]:

1. The cross-layer interoperability (also known as heterogeneity), which is defined as the orchestration of the Open System Interconnect (OSI) layers in a seamless and burden-less way. Recently, this issue is solved by using two emerging virtualization technologies; namely, Software-Defined Radio (SDR) and Software-Defined Networking (SDN) [7]. On one hand, SDN is a network architecture based on the separation of the control plane and the data plane which makes it possible to obtain a directly programmable network. This separation unifies the control plane over all kinds of network devices, making the network configured and optimized based on the network status [28]. On the other hand, by providing a higher degree of flexibility in designing wireless interfaces competent enough to support multiple communications technologies, SDR [39] can be a solution for addressing the interoperability problems. Akyildiz *et al.* [53] mentioned that a vast number of devices in IoTs constitute a fundamental challenge to the ubiquitous information transmissions through the backbone networks. The heterogeneity of IoT devices and the hardware-based, inflexible cellular architectures possess even greater challenges to enable efficient communication. Hence, they offered an architecture on wireless SDN and proposed software-defined gateways that jointly optimize cross-layer communication functionalities between heterogeneous IoT devices and cellular systems.
2. The cross-system interoperability, which is defined as maintaining the continuous operation of the networks and the systems within the existence of different system architectures. E.g., part of an IIoT might consist of LoRa-based end-devices and the other part might be dominated by Nb-IoT end-devices. This issue is may be solved by using semantic system models that provide a deeper understanding of the raw sensor data, by enabling AI-based machines to make decisions based on simple rules [25]. The design of gateways for interoperability can also be achieved by defining comprehensive centralized metadata and defining abstract models out of them that can be automatically generated for multiple programming languages.

2.4 Sensor Data Streaming,

Data streaming is a novel and powerful tool for industrial networks, which will enhance capabilities of the data analysis teams (especially in cyber-security). However, there are several challenges that needs to be accounted for [3]:

Plan for scalability: Seeks an answer for how much granularity and processing is needed on the data so that the network scales reasonably and can be kept under control.

Plan for data durability: Seeks ways of long-term data protection while storing the data, so that they do not suffer from bit fault, degradation or other corruption. Rather than focusing on hardware redundancy, durability is more concerned with data redundancy so that data is never lost or compromised.

Plan for fault tolerance: Incorporating fault tolerance in both the storage and processing layers is also important.

2.5 Safety

IIoT sensors and devices play critical a role in industrial (manufacturing, transportation, etc.) safety. IIoT leverages low cost and low power devices to implement energy-efficient safety protocols while improving productivity. The primary focus of safety is on internal risk-based problems in the manufacturing pipeline to prevent broad-spectrum issues like everyday work related to small accidents and as well as major disasters such as nuclear accidents and incidents. Many industrial facilities utilize multi-layer safety protocols[8], such as process control monitoring and alarms, Safety Instrument Systems (SIS), physical protection systems, emergency response (local / external) systems.

Alarms constitute a mechanism to inform a certain threshold is breached for a monitored event. Monitoring: Monitoring[9] is one of the most critical components of safety protocols. Most of the production pipelines require real-time continuous monitoring. Sensors generate real-time analog/digital data and transmit them to a control unit to analyze and perform activities based on the monitoring value. Latency is an important problem in monitoring systems. Moreover, physical monitoring is also important like corrosion/erosion monitoring to figure out detrimental effects. Physical protection system involves following equipment: Pressure-relief valves, rupture discs, stream traps, electrical switch-gear, eyewash stations, safety showers, etc.

Recent trends in IIoT safety systems are as follows:

- *Computer Vision (CV):* The CV technology is being used for anomaly detection (in detecting physical structural or machinery problems) along with other IIoT sensors.
- *Wearable electronics:* Involves monitoring of Vital life-signal measurements, fall detection, electric shock detection, etc.

[9] More information available at: https://www.isa.org/intech/201804web/

- *IIoT security to ensure safety:* Cybersecurity is also a major concern for safety systems. All IIoT devices should be cybersecurity proof during the operational phase. An external party like competitors or hackers should not halt or cause physical destruction of a production facility or a pipeline. For instance, in June 2010, Iran suffered a cyber-attack in a nuclear facility located at Natanz via Stuxnet cyber-worm [29]. Another major incident was on 31 March 2015, Iranian hackers took down the Turkish power grid which caused a massive power outage of 12 hours in 44 out of 81 cities. Approximately 40 million people were affected during this outage [32].
- *Robotics:* AI-driven software technologies are embedded to achieve autonomous decision-making machinery called robots. These devices can operate where a human can not. Such as in nuclear power plant or under the sea, etc.

2.6 Security and privacy

Interfacing the smart factories with Cyber-Physical Systems (CPSs) and IIoT improves the intelligence of the infrastructures, yet introduces cyber-security vulnerabilities which may lead to critical problems such as system failures, privacy violations and/or data integrity breaches. As the privacy of the citizens is becoming into prominence, especially in the EU with the GDPR act, privacy bearing information of IIoT users such as Personal Identifying Information (PII), need to be treated well, so that they will be kept confidential [9].

As thoroughly discussed in [12], many IoT networks do not even possess basic security elements. On average, these are the cyber-security analysis of today's COTS (Commercial off the shelf) IoT products[10]: 25 vulnerabilities are detected per device, 60% has vulnerable firmware's and user interfaces, 70% do not encrypt any communications at all, and 80% fails to request password for authentication that has a secure secure length. Henceforth, there are two main methods to fight against intrusions and cyber-attacks against IIoT. One of them is allowing intrusions to happen and then detecting them via Intrusion Detection Systems (IDS) as in [5]. The other one is prevention of attacks by means of authentication, authorization and access control.

2.6.1 Intrusion Detection

In practice, IDSs are installed and on demand from every aspects of technological life, from corporate to universities where IT department exists. Especially, following are the topics of our interest:

- Industrial network and IIoT security

[10] by COTS vendors such as ABB, Arm, Bosch, Huawei, Intel, Siemens, Netvox, etc.

- Smart grid security
- Critical infrastructure security
- Smart city/factory/home security

The main distinction among the anomaly-based IDS and misuse/specification-based IDS is, anomaly detection-based systems can detect any kind of bad behavior (theoretically) on the fly but misuse/specification detection-based systems can only recognize previously known bad behaviors (signatures) [11].

The reasoning behind this is as follows: Misuse/specification detection-based systems are designed in a way to match previously modelled attack vectors and rules. If an incidence can be categorized in these, then it is called as an attack. Otherwise, it is called as normal behavior. These systems work very well with the defined and categorized attacks up until the time of implementation. However, they are quite useless in the case of new attack vectors that can not be specified with the old ones. For these situations, anomaly detection-based IDS is suggested. Hence it is easy to model the normal state than the abnormal one, it is modelled to specify the good (or normal) phase of the system behaviors. Therefore, it is logically opposite of the misuse/specification detection-based systems in which the attack signatures and vectors are directly identified. So, anomaly detection-based IDS uses a kind of live reasoning algorithm, meaning that the decision on a future incident might differ from a past one, even if they are the same events.

Besides, as mentioned deeply in above discussions, they are easier to setup, as modelling the normal operating conditions (phase) of the network is easier than specifically identifying each attack vector. Therefore, the next subsection is dedicated for that.

Each of these approaches is then sub-classified into various methods. The interested reader is referred to [10] and also Chapter-6 of this book, for a more detailed discussion.

2.6.2 Intrusion Prevention

It is referred to as taking all necessary actions required in order to prevent intrusions. It might be analogically similar to theft protection systems in real-life security applications such as installment of advanced door locks, infrared detectors, etc. Some of the methods to be mentioned are as follows but not limited to [8]: Authentication, authorization, access control, ciphering (encryption/decryption), hashing, etc.

2.6.3 Runtime Security Monitoring

The runtime security monitor operates on the state of the industrial system and observes anomalous behavior that occurs either by failures and operational errors or security threats by adversaries. To guarantee its successful operation the runtime security monitor requires knowledge about the system as well as the trusted states, operation conditions as well as the system output information i.e. what the system

produces. The objective of the runtime security monitor is to identify suspicious and anomalous indicators when i.e. the system stops producing what is intended. These are depicted as faults or security threats (Figure 7). Then, it acts proactively by replacing the system with a so-called "reversionary" system, that is performing the minimum industrial system functionality defined by the requirements. The reversionary system ensures redundancy and reliability.

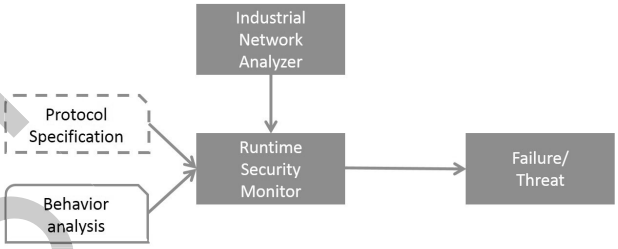

Fig. 7: The concept of runtime security monitoring.

The presence of a runtime security monitor should be always coupled with an industrial network analyzer (Figure 7), as the events and data that are exchanged in the industrial network are encoded in proprietary protocols and message formats. An example of a network analyzer is Wireshark [11] providing support for some industrial protocols such as Modbus and DNP3 as well as libraries and extensions for other proprietary protocols (e.g. Siemens Step 7).

3 Future trends in industrial networks

Here, it is worth mentioning that, WSN represents an earlier version of sensors (sensing devices) that have sensing, communication and networking capabilities to set up a network and send sensed information to the data sink throughout the network. IoT is evolved from WSN notion with the need of reaching those sensors and actuators via global network of Internet. IIoT is a subset of the IoT especially devised to support industrial automation; therefore possess most of the characteristics of generic IoT networks. Hence, the OSI network structure of an IIoT (and IoT) consists of 5 layers as described in [36]: Physical, Data-Link (MAC), Network, Transport, and Application. It should be noted that Session and Presentation layers of the traditional OSI network model are all considered in the Application layer of IIoT (and IoT).

It is expected the future industrial networks to include more wireless technologies such as the emerging ones 5G, LoRa, etc., in order to support the *automation* notion of Industry 4.0 by enabling remote command, control and monitoring of the sensors and actuators on the factory floor. It might be expected these wireless technologies

[11] https://www.wireshark.org/

to enable Internet connectivity to industrial networks, which would eventually cause both terms, industrial networks and IIoT to be federated under same term. Extensively usage of automation over the industrial domain will also necessitate an increased number of remote sensor and actuator installations, resulting in the creation of a vast amount of data, called big data. Besides, as most industrial networks are mission-critical, the security and safety of these networks will be important. Therefore, filtering information from the big data will be one of the most challenging aspects of the industrial networks, as a timely response is really critical in the industrial environments. One recent solution to this dilemma (on where to process the data) is, pushing cloud/server tasks/missions towards the edge of the network, from which the term edge/fog computing emerged. Due to CISCO, *fog computing* extends and complements the *cloud computing* with the concept of smart devices which can work on the edge of the network [15].

Authors of this chapter envision that the *data streaming* paradigm will be an enabling technology for the *fog computing* notion of the CISCO, and therefore play a critical role at the future industrial networks. Data streaming will help fog computing, by processing the data closer to the source, decreasing the amount of traffic created, and also reducing the overall response time for the queries. This will be especially beneficial in detecting intrusions. According to the authors' insight, in the near future, the data streaming paradigm (described thoroughly in Chapter-6 of this book) along with a distributed architecture will be adopted by industrial networks.

3.1 Industry 4.0

The industrial revolution started with the mechanization (water/steam power) of the production pipelines. Steam engines are used in production and transportation. The second transformation has happened once the steam power is replaced with electricity. Breakthrough in transistor and electric circuits technology produced computers which triggered the third industrial revolution. Mass device connectivity nudges the fourth industrial revolution which is enhanced with smart devices and advanced artificial intelligence algorithms like machine learning. The main components of Industry 4.0 and IoT are as follows: Cloud, cyber-security, IoT, system integration, simulation, autonomous robots, Big Data, augmented reality, and additive manufacturing. The history of Industry 4.0 is well explained in Chapter 4.2. Briefly, the focus of Industry 4.0 is to do optimization in computerization and eliminate human involvement in the decision-making process to eliminate human errors and improve efficiency. This can be done with smart devices (monitoring) and automation.

3.2 Industrial Site Indoor and Outdoor Lighting

IIoT-based systems, also in conjunction with Light Emitting Diode (LED) luminaires, intelligent controls by using RF and Power over Ethernet (PoE) technology, is enabling significant advancements in next-generation building lighting systems for commercial and industrial buildings as well as street lighting. Especially, Chapter-3 of this book is dedicated for this specific application area (commercial and industrial building lighting and in street lighting) of IIoT networks to industrial domain. The technical, economic, and market aspects of this group of technologies are discussed and presented thoroughly in Chapter-3.

3.3 Smart Grid Systems

Smart grid systems not only consist of an electricity distribution network, but also a coupled information network on top. These systems mostly consist of smart meters, nowadays of which mostly composed of IoT devices, hence might be a good example of an IIoT network.

In most cases, bottlenecks occur in the upstream path of the networks. Intrusion detection solution (especially related to the data streaming) discussed in this book (Chapter-6), aims at pushing the data analysis towards peripheral devices instead of gathering data centrally. This kind of solution is offered to provide a fast response by removing the necessity of the round-trip messages in between the smart meter and the server. On the other hand, data streaming is also beneficial to the overall network performance due to the decreased load in the aforementioned bottlenecks.

3.4 NarrowBand IoT (NB-IoT)

Apart from the 3GPP existing 4G and ongoing 5G standards, 3GPP is also providing standards for cellular Low Power Wide Area Network (LPWAN) communication mechanisms and functionalities targeted in low-end devices, such as the ones used in the IoT devices. A representative example in this category is *Narrowband-IoT (NB-IoT)* or often referred to as LTE-M2 [45].

The NB-IoT provides low energy consumption, small volumes of data and transmission over large distances or deep within buildings. It is based on release 13 of 3GPP and operates at even lower bandwidths (180 kHz/channel) and lower data rates (20 kbps) in the licensed LTE spectrum. Mobility is sacrificed in favor of better indoor coverage and support for larger number of devices. NB-IoT is managed by cellular operators with expected costs and regulations on access to this network.

The 3GPP offers three scenarios for LPWAN deployment in NB-IoT (illustrated in Figure 8), which are namely In-Band, Guard-Band and Standalone. Specifically, In Band makes use of the same resource block in the LTE carrier of the existing

LTE network. Furthermore, guard-band deployment uses the unused blocks within the LTE carrier guard band and standalone deployment utilizes new bandwidth in comparison to existing technologies (e.g. GSM, LTE).

Fig. 8: Scenarios for LPWAN deployment in NB-IoT.

3.5 LoRa and LoRaWAN

LoRa (Long-Range) is a subclass of LPWAN or Low-Power Wide-Area (LPWA), which is a new class of communications technology devised for IoT network implementations. LoRa describes physical layer communications technology, which is a Chirp Spread Spectrum (CSS) to spread the signal in the frequency band in order to get resistance against wireless signal interference and fading [14]. LoRaWAN describes all upper layer OSI stacks to enable seamless packet transmission. Apart from its earlier version (v1.0), new version (v1.1) of LoRAWAN is known to be secure against most of the cyber-attacks as shown in [13] and [27].

LoRaWAN [38] is an openly defined network protocol that manages communication between gateways and end-devices with the following features: (1) establishing encryption keys for application payloads and network traffic, (2) device to gateway pairing assignments, and (3) channel, power and data rate selection. The devices in LoRaWAN can be of three types:

1. bi-directional end-devices with downlink followed by uplink, as for example sensor end devices,
2. bi-directional end-devices with transmission slots scheduled for downlink, as in the case of actuators, and
3. always-on bi-directional devices, which is intended for low-resource devices to ensure low-latency such as gateways or servers.

A reference view of the LoRaWAN architecture is provided in Figure 9. The figure illustrates the communication between the LoRaWAN Server, the gateways as well

as the end devices. The gateways are responsible for maintaining radio connectivity as well as may act as transparent bridge on the network. Furthermore, they ensure seamless network upgrade. Additionally, the LoRaWAN Server is responsible for maintaining association with end node, configuring data rates, removing duplicates and the handling security and access control interfaces with applications. Finally, the LoRaWAN end device in the system has a network communication and application encryption key. All packets are transparently sent from gateways to a LoRaWAN server without any local decryption to limit the potential risk of compromised clients and gateways (Figure 9).

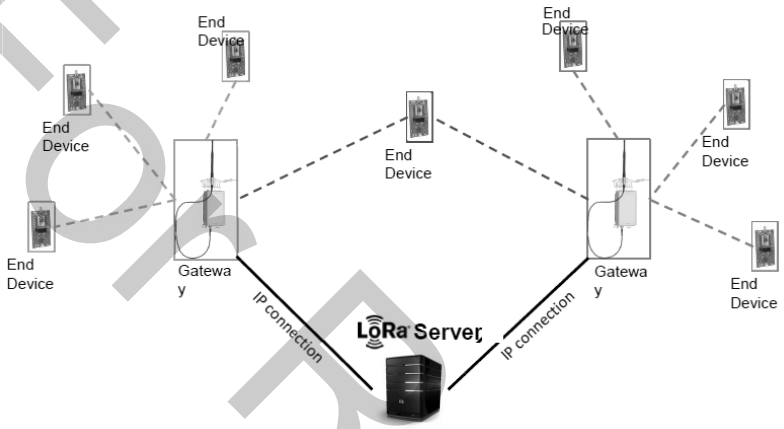

Fig. 9: Reference architecture in LoRa

3.5.1 LTE (Long-Term Evolution)) for machine type communications

LTE-MTC or LTE-M is an LPWA technology standard based on 3GPP's Release 13 specification. It specifically refers to LTE Cat M1, suitable for the IoT. Even though both NB-IoT and LTE-M use LTE and aims in enabling low-power communication in IoT devices, their main differences are in terms of throughput, mobility, power, latency, and cost.

Table 1 provides insight on the difference in technology characteristics for NB-IoT and LTE-M. The main differences are found in terms of mobility as well as the technology design. Specifically, NB-IoT does not provide mobility when change from one cell to another and the User Equipment (UE) have to perform idle rejoin. This introduces a power penalty for moving devices). NB-IoT is good for sending small and subsequent messages, whereas LTE-M is used to send sequences of messages, such as data streams. Additionally, LTE-M and NB-IoT have also a difference in the power saving mode that the support. In particular, LTE-M supports several power saving modes (e.g: deep-sleep or wake-up only periodically while connected), whereas

NB-IoT supports the Extended Discontinuous Reception (eDRX) [35] mode, which allows reduced power consumption for devices that are awake and remain connected.

Technology characteristics	NB-IoT	LTE-M
Bandwidth	<250 kbps (Half Duplex)	384 kbps-1 Mbps (Half or Full Duplex)
Coverage	20 dB	15 dB
Mobility	No	Yes
Designed for	Message-Based Communication	IP-Based Communication
Power saving	eDRX [35]	Deep-sleep/periodic wake-up

Table 1: NB-IoT and LTE-M differences

3.6 5G

Wireless cellular technology in mobile communication has reached the 5G (Fifth Generation - 2019) milestone. 1G (First generation) cellular networks designed with analog technology in 1979. Whereas, 2G (Second generation - 1991, CDMA (Code-division multiple access), GSM (Global System for Mobile Communications), TDMA(time division multiple access)) cellular networks are digital which supports messaging like SMS and voicemail. 3G (Third generation - 2001, EVDO (Evolution-Data Optimized), HSPA (Evolved High-Speed Packet Access), UMTS (Universal Mobile Telecommunications System)) has faster data transfer rate $\sim 144 kbit/s$. which enables GPS and mobile web technologies. 4G (Fourth generation - 2009, WiMAX (Worldwide Interoperability for Microwave Access), LTE (Long-Term Evolution)) broadband utilizes IP and packet-switched technology to handle data rates of $\sim 100 Mbit/s$. 4G can be considered as the rise of the mobile Internet. 4G cellular networks can support the following applications: IP telephony, video conference, and streaming. 5G is the latest wireless technology that provides the fastest connectivity (download and upload speed). It has also following advantages: reliable connection, high-quality voice and video transmission, along with a support of increased number of connected (IoT) devices. 5G networks can be configured with the following options: Low-Band (operates in frequencies below 1GHz), Mid-Band (in the 1-10GHz range), High-Band (in the 20-100GHz range).

5G will affect our daily communication and mobile usage habits as well as reshape the industry. For instance, due to its video streaming capacity, it will change the way of journalism, etc. Now a single journalist can be deployed on the field with a 5G connected camera to do interviews instead of a news crew with heavy equipment requirements. Moreover, 5G will contribute to adopting advanced technologies such as self-driving cars and smart things (home, factory, city, etc.).

Figure 10 illustrates an architectural view of 5G communication. The figure divides the network into two parts, the user data part (also known as the user plane)

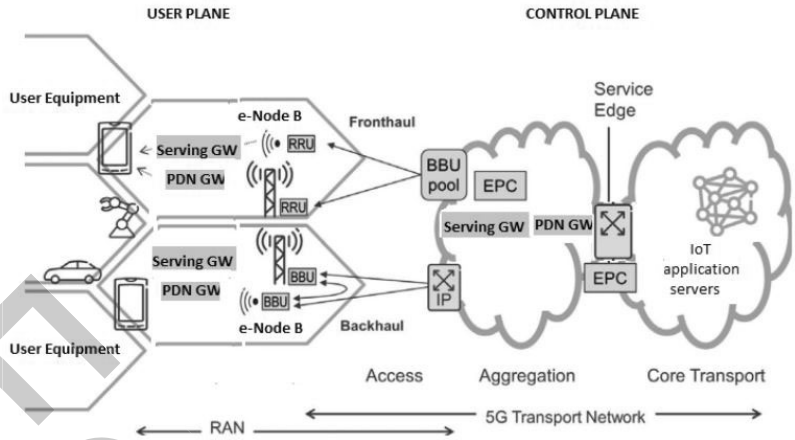

Fig. 10: 5G reference architecture

and the signaling part (also known as the control plane). This separates their concerns as well as makes the scaling independent. The control plane is supported by the 5G transport network and the latter by the Radio Access Network (RAN). Additionally, the former is further divided into fronthaul and backhaul packet networks. Backhaul is the linkage between a base station and the core wired network, and is often fiber or coax, and in some cases broadband, proprietary wireless links. In most cases the backhaul network is supported by wired communications to enable less communication latency. The front hall network provides the connection between the cell tower radio itself (Radio Head or RH) and the mobile network control backbone (the Baseband Unit or BBU) and CPRI is a well-known standard for this interconnection. The control plane includes additionally the communication with the IoT application servers and the EPC. Furthermore, the EPC comprises by two gateways the Serving and the Packet Data Network (PDN) gateway, serving as the Control Plane for the network. The former is responsible for routing the incoming and outgoing IP packets and the latter serves as a connection point between the EPC and the external IP networks, called as PDN. The PDN gateway routes packets to and from the PDNs and performs various functions such as IP address / IP prefix allocation or policy control and charging. The Serving gateway is logically connected to the PDN gateway and even though 3GPP describes them independently, in practice they may be combined in a single hardware by network vendors. Packets in 5G communication are exchanged between the cloud (application servers of Figure 10) and the control plane. The control plane is afterwards using the standard TCP, UDP and IP protocols to exchange packets with the user plane. The BBU is used to form the evolved (Evolved Node B (e-NodeB) in Figure 10) the main point responsible for the transmission/reception of IP packets to/from the control plane. The BBU also performs packet demodulation as well as amplification to transmit them to the User Equipment (UE), which denotes the end devices used for communication.

Connectivity between the user equipment UE and the core network is provided by the E-UTRAN. The E-UTRAN is a collective term for the network and equipment that connects mobile handsets to the public telephone network or the Internet. The user plane contains the e-NodeB and UE consists of three sub-layers: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC). In the same figure we can also observe data exchange through UDP between the UE and the Serving gateway, which serves as a relay for the EPC.

3.7 Cloud, Edge, and Fog computing

In *Cloud Computing*, collected data is streamed to a central server and data is processed on a powerful central server which is usually far away from the data source. Cloud technology provides various services like IaaS (Infrastructure as a Service), PaaS (Software as a Service) and SaaS (Software as a Service). Usually, cloud computing provides highly scalable and almost unlimited storage for business solutions. It also provides really high processing power when compared to the Fog/Edge computing. High latency, possible downtime, and security could be chronic issues in cloud computing for IoT.

Edge Computing brings processing power closer to where the data is being generated. It does not send generated data to a central location directly. The data can be processed on the device where the sensors are connected or a gateway device in close proximity. *Fog Computing* is very similar to the edge computing, as it can be considered as an extension which sets standards for edge computing and specifies how edge computing should work.

Fog/Edge: Computing and storage systems are located closer to the edge to eliminate latency. Fog/Edge computing can be integrated with cloud computing if needed. They can work as an extension of cloud computing. The concept of Fog/Edge is harmoniously compatible with IoT. The main goal here is to reduce the amount of sent data to reduce latency between the nodes to improve the overall system response time. Since the data is not transferred to a central location, it is considered more secure. Autonomous vehicles, like self-driving cars, need instant decision to avoid collusion or detecting pedestrians so the sensor or camera data needs to be analyzed on the vehicle (edge) to eliminate the latency or preserve availability. If there is a data link between the vehicle and a remote location, there is always a possibility of a breach or hijacking of the vehicle. It is known[12] that the Iranian cyberwarfare unit once successfully hijacked interrupting the data-link on RQ-170 Sentinel UAV manufactured by Lockheed Martin on December 5, 2011.

In avionics, many sensors are embedded on the aircrafts such as pressure, speed, engine, position, load, fuel, position, torque, steering, engine, gravity, etc. Considering commercial airlines to haul passengers we can say that data needs to be collected

[12] More information available at:
https://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident

from airplane sensors and needs to converted from analog to digital to inform the pilots for decisions[13]. This example shows how the Fog/Edge paradigm is used in commercial airlines since the data is processed on aircraft. However, considering Unmanned Aerial Vehicles (UAV) could be a good fit in cloud computing since UAV is operated from a central location based on the streamed sensor data from UAV sensors. Another example of cloud computing could be from space crafts and orbital satellites. Due to the hardware constraints, collected data needs to be streamed to earth for further processing and analysis.

Table 2: Comparison of cloud, fog and edge computing concepts [15].

Feature	**Cloud computing**	**Fog/Edge computing**
Access	Wired or wireless	Wireless
Access to the service	Through server	At the edge device
Availability	Mostly available	Mostly volatile
Bandwidth usage	High	Low
Capacity - Computing	Higher	Lower
Capacity - Storage	Higher	Lower
Connectivity	Internet	Many protocols (Fig 5)
Content distributed to	Edge device	Anywhere
Content generator	Man made	Sensor made
Content generation at	Central server	Edge device
Control	Centralized	Distributed
Data analysis	Long term	Instant / Short term
Data processing	Far from data source	Closer to data source
Latency	High	Minor
Location of resources (i.e. processing and storage)	Center	Edge
Scalability	High	Low
Security	Weaker	Stronger
Mobility	Limited	Supported
Number of users	Millions	Billions
Virtual infrastructure location	Enterprise server	Enterprise/User devices

3.8 Data Streaming

Data streaming is related with time series and optimal for detecting patterns over time. For instance, tracking and recording the length of a web session sent for a client side. Most of the IoT data is also well-suited to data streaming paradigm. Applications like security cameras, traffic sensors, crowd sensors, health sensors, activity logs, and transaction logs are all good examples for data streaming.

[13] More information available at:
https://www.azosensors.com/article.aspx?ArticleID=1614

Data streaming allows analyzing data in real time and gives insights into a wide range of activities, such as metering, server activity, geolocation of devices, or website clicks; by using some advanced computing such as real-time aggregation and correlation, filtering, or sampling. For example, in a smart grid network scenario, the AMI can monitor instant and cumulative throughput of the users and generates alerts when certain thresholds are reached [3].

As discussed earlier, with the wast amount of data (TBs) being created by the collection of AMI meters in a smart grid system, bottlenecks will occur in processing and storing this enormous big data. The solution to this problem might be processing the data closer to the source (smart meters) as much as possible. Data streaming algorithms employed at the edge of the network can help in this manner.

3.8.1 Data Handling and Computing in Information Systems

This sub-section overviews data handling and computing. There are two types of data handling for information systems in general [41]:

- **One-time query:** Traditional Data-Base Management Systems (DBMSs) have been used for decades to manage data. The primary goal of DBMSs is to store data in a form of persistent data-set and then run one-time queries over it.
- **Continuous query:** Data stream processing is one pass analysis over the data on the fly. In contrast to DBMSs, stream processing employs continuous queries which are queries that are issued once and continuously run over the flow of tuples (new data records).

As shown in the Fig. 11, stream processing can run the query immediately after a new tuple arrives which in turn enables online analysis. Although DBMS is useful in applications where the updates of the database are relatively infrequent, it is inefficient for modern high-volume (data size) / high-velocity (the speed of data generation or arrival) / high-veracity (discrepancy) / high-variety (various data structures) data-driven applications where Big Data (4 V's) is being generated. At that point, continuous query will be an efficient solution to be used for Big Data generating systems.

Authors of this chapter projects that, the traditional technique of *one-time query* will be replaced by the *continuous query* technique in the near future for the industrial networks, as it will be more beneficial not only for the network provider but also for the industrial users.

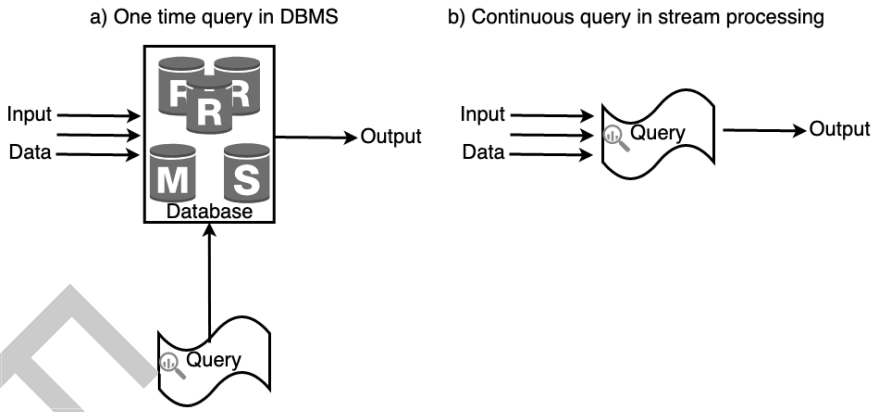

Fig. 11: Data handling in information systems. Here, *R* represents replica or reporting database (DB) instances, *S* represents standby DB, and *M* represents the active DB.

3.8.2 How the data is processed?

- **Data fusion:** Data fusion is the process of integrating data destined from multiple sources to produce more consistent, accurate, and useful information.
- **Sensor fusion:** A specific case of data fusion, in which data resourced from many sensors are incorporated into a meaningful and useful form.
- **Data streaming:** Continuously generated data by different sources should be processed incrementally using *data stream processing techniques* without having access to all of the data being produced or transmitted. It is usually used in the context of big data in which it is generated by various sources at high speed.

3.8.3 Data Streaming Tools

As the data streaming concept grows exponentially, a number of tools have already gained popularity among implementer and coders [3]:

Amazon Kinesis Firehose: It is a managed, scalable, cloud-based service which allows real-time processing of large data streams.

Apache Kafka: It is a distributed publish-subscribe messaging system which integrates applications and data streams all-together.

Apache Flink: It is a Stream Processing Engine (SPE) which works on streaming data flow and provides facilities for distributed computation over data streams.

Apache Storm: It is a distributed real-time computation system. Storm is used for distributed machine learning, real-time analytics, and numerous other cases, specifically with high data velocity and veracity.

3.9 Digital Twin

As discussed in [34], IIoT leads to the Industry 4.0, by improving the factory environments by enhancing the user interactions in between the machines and operators. For instance, *Digital Twin* is one of the most intriguing technological developments that will shape tomorrow's industrial environments.

> Especially, as foreseen in [50], Product Life-cycle Management (PLM) of industrial environments will benefit from *Digital Twin* concept. It will not only ease the way of overall production monitoring, but also enhance the predictability of the events of malfunctions, bottlenecks in the production line, etc. Up-to date, PLM is very time-consuming and challenging in terms of manufacturing, sustainability, efficiency, intelligence, and service phases of the product design. The digital twin will enable manufacturing operators to have a digital replica of all of their production lines through the entire product life-cycle. Hundreds of sensors will be placed in the manufacturing process environment, to collect data from various dimensions such as ambient conditions, operational characteristics of the machines along with their working efficiency on the job they are performing. All this data is continuously communicating and collected by the digital twin [52].

In [1], it is clearly mentioned that: "As the technology trend of industrial IoT increases, digital twin technology is more significant now than ever before." Owing to the improvements in IoT and especially on IIoT, digital twin became very affordable and has an enormous potential to realize the future of the manufacturing industry. The benefit provided to engineers is the fact that real-world products can be designed and tested virtually by the digital twin. Product maintenance and management can be drastically improved if the real 'thing' of the production environment to be replaced by the digital twin within real-time precision.

IIoT is en enabling technology for Digital Twin by providing the functionalities of remote sensing of the sensors and remote controlling of the actuators. As shown in Fig. 12, IIoT provides the means of communications to provide the link in between the real factory machinery and the digital twin equivalent at the command center of the factory.

4 Enabling Technologies for Industrial Networks

This section[14]thoroughly describes most (if not all) of the enabling technologies for industrial and IIoT networks, namely as follows: Ethernet, IP, PROFINET, CBA, Modbus, TCP, CANopen, NMT, PDO, SDO, Ethernet Powerlink, EtherCAT, Ether-

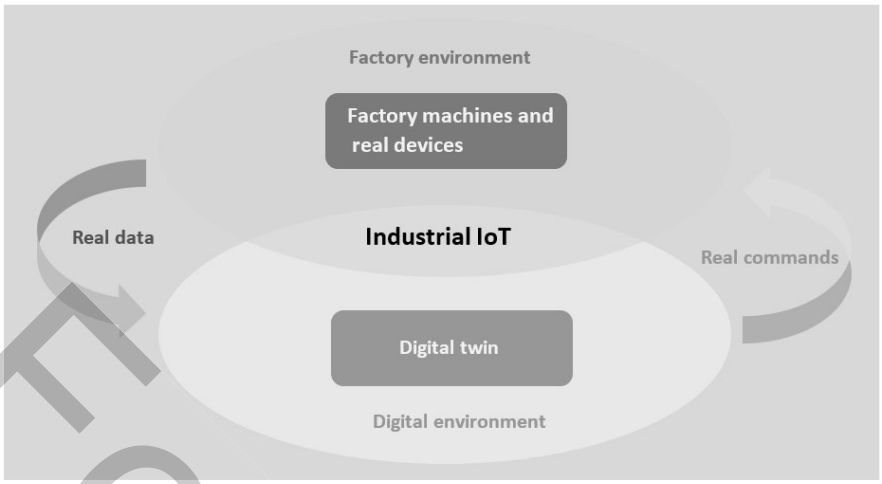

Fig. 12: IIoT and Digital Twin concept for tomorrow's factory environments.

CAT, V2V, V2X, V2I,I2V, V2P, P2V, V2N, N2V, I2N, N2I, DLMS/COSEM, DNP3, ZigBee, and BACnet.

4.1 Recent Developments in Industrial Networks for Industry 4.0

Industrial automation systems are used in manufacturing, quality control and material handling processes. General purpose controllers for industrial processes include Programmable Logic Controller (PLC) devices as well as sensors and actuators. The exchanged data in industrial processes are stored in powerful computers, such as servers. The main concern in such systems is to assure real-time performance as well as efficiency in terms of resource usage, such as the energy or memory consumption. Typical examples of such systems are distributed control systems or safety critical systems.

The main technologies used nowadays in industrial automation systems are called fieldbus protocols. They provide a digital communication link between control devices (input or output), which serves as a Local Area Network (LAN). Fieldbus technologies offer several characteristics, such as installation flexibility, maintainability (monitoring and maintenance are handled through the network) and most of all configurability. The latter provides a high degree of parameterization in the control devices, thus making them reasonably intelligent. The most common solutions in the family of fieldbus protocols rely on the Real-Time or Industrial Ethernet [24]. Real-Time Ethernet is using the standard Ethernet communication and apply

[14] This section is partially extracted from the unpublished part of the Ph.D. thesis of Dr. Lekidis [37], the full version of which is available at: https://tel.archives-ouvertes.fr/tel-01261936v2/document

modifications to extend it with real-time capabilities. Currently, a lot of Real-Time Ethernet solutions are in use, but only some of them are known due to their technical aspects and standardization status. Many of these solutions are defined in the IEC 61784-Part 1 [48] and IEC 61784-Part 2 [49] international standards for fieldbus communication and rely mainly on the master/slave architecture. In such an architecture a particular device manages the network and has uniformal control over the other devices.

The Real-Time Ethernet that employ a master/slave architecture are classified into three categories according to the implementation of the slave devices in the system. We hereby present these categories by evenly giving characteristic examples of technologies that are mainly described by the IEC 61784-Part 1 and IEC 61784-Part 2 international standards for each one of them. Moreover, for solutions that are not included in these standards, supporting material is provided.

The first category is using the TCP/IP protocol stack and hardware, such as the standard Ethernet controller as well as Ethernet switches. However, it does not provide guarantees for real-time performance as the communication latencies deriving from the use of switches as well as of the best-effort delivery service are unpredictable and result in an average data rate of 100 ms. Typical technology variants belonging in this category include Ethernet/IP, PROFINET Component Based Automation (CBA) and Modbus/TCP. The second category uses the same hardware, but employs an additional timing layer in the third layer (Internet) of the TCP/IP stack, in order to control access to the medium. Technology variants belonging in this category include PROFINET Real-Time (RT) and Ethernet POWERLINK (EPL) (see Section 4.2.8). An important feature of this category is that it provides better real-time performance (average data rate below 10 ms), which can be additionally ameliorated as some of the related technologies are also deployed using Ethernet hubs (e.g. Ethernet POWERLINK). Finally, the third category aims on achieving the best possible real-time performance for the most demanding class of applications. Nevertheless, this is not feasible without specific modifications on the underlying hardware. These modifications depend on the technology and can either concern the Ethernet controller or the Ethernet switches. Technologies related to this category include PROFINET Isochronous Real Time (IRT), SERCOS III, EtherCAT and TTEthernet [51]. The selection of the category as well as the specific master/slave solution for an application depends on its requirements and needs.

Even though Real-Time Ethernet technologies are widely used for industrial automation systems, application development is still challenging, due to their low level complexity as well as their high expertise needed for their configuration. Therefore, a higher layer of abstraction is required, which is typically found in application-layer protocols. An increasingly popular application-layer fieldbus protocol is CANopen (see Section 4.2.1), as it provides a vast variety of communication mechanisms, such as time or event-driven, synchronous or asynchronous as well as additional support for time synchronization and network management. Furthermore, it offers a high-degree of configuration flexibility, requires limited resources and has therefore been deployed on many existing embedded devices.

4.2 Industrial Automation Protocols

4.2.1 CANopen

CANopen [20] is an increasingly popular application layer protocol, belonging to the family of fieldbus protocols for networked embedded systems. Its main attributes are the vast variety of communication mechanisms, such as time or event-driven, synchronous or asynchronous as well as the support for time synchronization and network management mechanisms. Additionally, it provides a high-degree of configuration flexibility and requires limited resources. CANopen uses a master/slave architecture for management services, but concurrently allows the utilization of the client/server communication model for configuration services as well as the the producer/consumer model for real-time communication services. A comprehensive introduction to the protocol can be found in [42]. Unlike other fieldbus protocols it does not require a single master controlling all the network communication. Instead a CANopen system is specified by a set of devices (Figure 13), which in turn use a set of profiles, in order to define the device-specific functionality along with all the supported communication mechanisms. The communication profile defines all the services that can be used for communication and the device profile how the device-specific functionality is made accessible. The communication profile is defined in the DS-301 standard [20], whereas the device profiles providing a detailed description on CANopen's usage for a particular application-domain, are defined in the DS-4xx standards [15]. If CANopen systems require configurations or data access mechanisms not covered by the standard communication profile, profile extensions can also be defined. These are called Frameworks and are found in the DS-3xx standards [15].

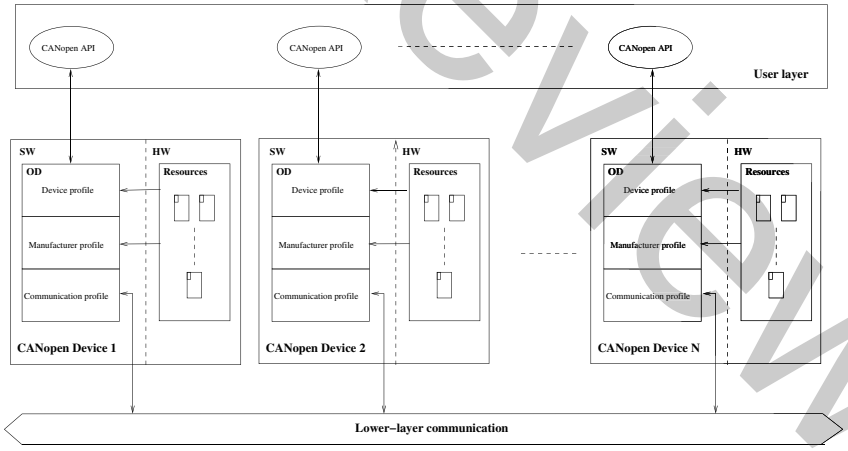

Fig. 13: Communication in a CANopen system

[15] http://www.can-cia.org/index.php?id=440

The protocol's communication mechanisms according to the DS-301 are specified by standard *Communication Objects (COB)*. All the COBs have their own priority and are transmitted through regular frames of the chosen lower-layer protocol. They are generally divided in the following main categories:

- *Network Management objects (NMT)*, used for the initialization, configuration and supervision of the network
- *Process Data Object (PDO)*, used for real-time critical data exchange
- *Service Data Object (SDO)*, used for service/configuration data exchange
- *Predefined objects*, specifying standard object that are included in every device. The featured objects in this category are:
  - *Synchronization object (SYNC)*, broadcasted periodically to offer synchronized communication as well as coordinate operations
  - *Timestamp object (TIME)*, broadcasted asynchronously to provide accurate clock synchronization using a common time reference
  - *Emergency object (EMCY)*, triggering interrupt-type notifications whenever device errors are detected

All the aforementioned objects are stored in a centralized repository, called Object Dictionary (OD), which holds all network-accessible data and is unique for every device. Commonly used to describe the behavior of a device, it supports up to 65536 objects addressed through a 16-bit index. The COBs are spread to distinct areas, defining communication, manufacturer and device specific parameters (Figure 13). The latter are left empty and are used by manufacturers, in order to provide their own device functionalities. Each OD entries has also an associated data type with a respective code, used to identify it. The supported data types along with their code are given by the following Table.

Index	Data type
1	BOOLEAN
2	INTEGER8
3	INTEGER16
4	INTEGER32
5	UNSIGNED8
6	UNSIGNED16
7	UNSIGNED32

Table 3: Frame fields in the CAN HW/Communication Model

Furthermore, every CANopen object in the OD has a dedicated type with respect to the information it stores. and each type has an associated code. In particular, the CANopen object may be either a variable (object code 7), an array (object code 8) or a record (object code 9). The difference between the array and the record is that the former contains sub-indexes of the same data type, whereas the latter of different data types.

The OD entries are described by electronically readable file formats, such that they are uniformly interpreted by configuration tools and monitors. According to the DS-306 standard [18] they are provided by the INI format files and termed as Electronic Data Sheet (EDS) files. These files provide a generic description of a device type. However, since CANopen allows parametrization according to manufacturer specifications, a specific file format exists and is defined as Device Configuration File (DCF). This file describes the configuration for a specific device. Nevertheless, EDS and DCF files have limitations on the validation and presentation of the data as well as require a specific editor. Therefore, new XML-based device descriptions were introduced according to the DS-311 standard [19]. These substitute the EDS with the XML Device Description (XDD) file format and the DCF with the XML Device Configuration (XDC) file format. A fragment of an XDC a device description is provided in 15. Currently, the protocol supports both device descriptions.

We hereby describe thoroughly the CANopen objects, according to the classification we have previously mentioned.

4.2.2 Network Management (NMT) Objects

The NMT objects are generally transmitted by devices, which act as an NMT master in CANopen. Upon the reception of such object a CANopen device is informed to transit in a different NMT state. Each NMT state supports specific communication mechanisms and objects of the CANopen protocol, related to the device functionality. The device can switch between three main state, named Pre-Operational, Operational and Stopped. In the Pre-Operational state a device can actively participate in all communication mechanisms related to SDO and Predefined object exchange. However, the main difference with the Operational state is that it doesn't support PDO object exchange. In the Operational state the device is fully operational and can perform all the functionalities that it was designed to do. The NMT master can also switch off the device by transmitting a dedicated NMT object. Accordingly, the device has to switch to the Stopped state stopping all communication, except from the support for the reception of NMT objects.

4.2.3 Process Data Objects (PDO)

The real-time data-oriented communication follows the producer/consumer model. It is used for the transmission of small amount of time critical data. PDOs can transfer up to 8 bytes (64 bits) of data per frame and are divided in two types: The transmit PDO (TPDO) denoting data transmission and the receive PDO (RPDO) denoting data reception. Therefore, a TPDO transmitted from a CANopen device is received as an RPDO in another device (Figure 14). Additionally, the supported scheduling modes are:

- *Event driven*, where the transmission is asynchronous and triggered by the occurrence of an object-specific event

- *Time driven*, where transmission is triggered periodically by an elapsed timer
- *Synchronous transmission*, triggered by the reception of the SYNC object, further divided in:
  - Periodic transmission within an OD-defined window (synchronous window), termed as *Cyclic PDO* transmission
  - Aperiodic transmission according to an application specific event, termed as *Acyclic PDO* transmission
- *Individual polling*, triggered by the reception of a remote request (see [17])

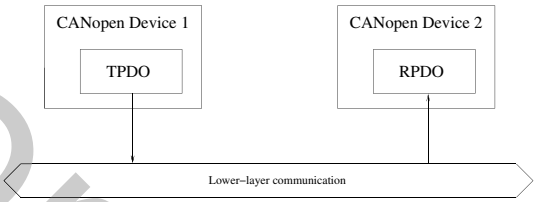

Fig. 14: PDO communication

Each PDO is described by two OD sub-objects: The Communication Parameter and Mapping Parameter. For a TPDO (e.g. OD entry 1800h and 1A00h respectively) the former indicates the way it is transmitted in the network and the latter the location of the OD entry/entries, which are mapped in the payload. On the contrary for a RPDO (e.g. OD entry 1400h and 1600h respectively) the former indicates how it is received from the network and the latter the decoding of the received payload and the OD entry/entries where the data is stored.

The Communication Parameter entry includes the *Communication Identifier (COB-ID)* of the specific PDO, the scheduling method, termed as *transmission type*, the *inhibit time* and the *event timer*. The inhibit time (expressed as a multiple of 100 μs) defines the shortest and the event timer (expressed as a multiple of 1 ms) the longest time duration between two consecutive transmissions of the same PDO.

The Mapping Parameter describes the structure of a PDO. It can be of two types, that is, static or dynamic. Static mapping in a device cannot be changed, whereas dynamic mapping can be configured at all times through an SDO.

In Table 4 we illustrate the sample configuration and mapping parameters of a TPDO for a CANopen device. They represent how analogue input data, obtained from temperature sensors, are described in the OD of the device. Accordingly, we also present the fragment, which describes them in an XDC format.

Example

The fragment of Figure 15 illustrates the TPDO configuration of Table 4 in an XDC format. We can observe that all the CANopen objects are defined inside the construct “CANopenObjectList". Moreover, for each object (“CANopenObject") dedicated

Index	Subindex	Description	Value
1800h (6144)	0	Number of entries	5
	1	COB-ID	641 + deviceID
	2	Transmission type	255
	3	Inhibit time (in ms)	1
	4	Reserved	-
	5	Event timer (in ms)	1000

Index	Subindex	Description	Value
1A00h (6656)	0	Number of entries	1
	1	1st object to be mapped	6400h (25600)/Subindex 1
6400h (25600)	0	Number of analogue inputs	n
	1	input 1 (in °C)	30.5
::::	::::	::::	::::
	n	input n (in °C)	23

Table 4: Example TPDO configuration and mapping parameters in the OD

elements are also defined to denote the sub-objects (“CANopenSubObject"). Every sub-object has an index (“subIndex"), a name (“name"), a specific object code (“objectType") and a type of data that it may contain (“dataType"). Additionally, a sub-object includes a default value (“defaultValue") and its specific value in the application (“actualValue"). Nevertheless, the latter is optional and if it not provided in the XDC file, it is accordingly set equal to the default value.

```
<ProfileBody>
        <ApplicationLayers>
        <CANopenObjectList>
        <CANopenObject index="1800" name="1st␣Transmit␣PDO␣Communication␣Parameter" objectType="9" subNumber="5">
                <CANopenSubObject subIndex="00" name="Number␣of␣entries" objectType="7" dataType="0005" lowLimit="0x02"
                highLimit="0x05" accessType="ro" defaultValue="5" PDOmapping="no" actualValue="5"/>
                <CANopenSubObject subIndex="01" name="COB-ID" objectType="7" dataType="0007" PDOmapping="no"
                uniqueIDRef="UID_PARAM_180001"/>
                <CANopenSubObject subIndex="02" name="Transmission␣Type" objectType="7" dataType="0005" PDOmapping="no"
                uniqueIDRef="UID_PARAM_180002"/>
                <CANopenSubObject subIndex="03" name="Inhibit␣Time" objectType="7" dataType="0006" PDOmapping="no"
                uniqueIDRef="UID_PARAM_180003"/>
                <CANopenSubObject subIndex="05" name="Event␣Timer" objectType="7" dataType="0006" PDOmapping="no"
                uniqueIDRef="UID_PARAM_180005"/>
        <CANopenObject>
        .....
        <CANopenObject index="1a00" name="1st␣Transmit␣PDO␣Mapping␣Parameter" objectType="9" subNumber="9">
                <CANopenSubObject subIndex="00" name="Number␣of␣entries" objectType="7" dataType="0005" accessType="ro"
                defaultValue="1" PDOmapping="no" actualValue="8"/>
                <CANopenSubObject subIndex="01" name="PDO␣Mapping␣Entry" objectType="7" dataType="0007" PDOmapping="no"
                uniqueIDRef="UID_PARAM_1a0001"/>
        </CANopenObject>
        <CANopenObject index="6400" name="Read␣Analog␣Input␣16-bit" objectType="8" subNumber="13">
                <CANopenSubObject subIndex="00" name="Number␣of␣elements" objectType="7" dataType="0005" accessType="ro"
                defaultValue="12" PDOmapping="no" actualValue="N"/>
                <CANopenSubObject subIndex="01" name="AnalogInput16_1" objectType="7" dataType="0003" PDOmapping="TPDO"
                uniqueIDRef="UID_PARAM_640101"/>
                .....
                <CANopenSubObject subIndex="01" name="AnalogInput16_N" objectType="7" dataType="0003" PDOmapping="TPDO"
                uniqueIDRef="UID_PARAM_64010N"/>
        <ApplicationLayers>
        .....
</ProfileBody>
```

Fig. 15: TPDO configuration in an XDC CANopen specification

4.2.4 Service Data Objects (SDO)

The service oriented communication follows the client/server model. It supports large, non-critical data transfers and uses three modes to allow peer-to-peer asynchronous communication through the use of virtual channels:

- *Expedited transfer*, where service data up to 4 bytes are transmitted in a single request/response pair.
- *Segmented transfer*, where service data are transmitted in a variable number of request/response pairs, termed as segments. In particular it consists of an initiation request/response followed by 8-byte request-response segments.
- *Block transfer*, optionally used for the transmission of large amounts of data as a sequence of blocks, where each one contains up to 127 segments.

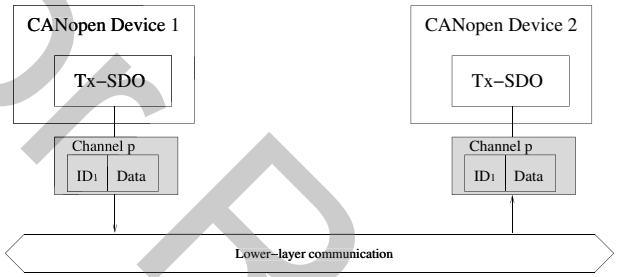

Fig. 16: SDO communication

A CANopen device can either receive or request an SDO, therefore these objects are not separated as the PDOs, instead they are distinguished according to their identifiers (Section 4.2.6). The communication is always initiated by a device defined as client in the network towards the server, nonetheless information is exchanged bidirectionally with two services: *Download* and *Upload*. The former is used when the client is attempting service data transmission to the server, whereas the latter when it is requesting data from the server. In both services the use of the virtual channel ensures that the received SDO is identical to the transmitted (Figure 16), unlike in PDO communication. Each request or receive SDO uses byte 0 as metadata, containing important information about the transmitted object and reducing the payload to seven bytes per frame. This byte includes the command specifier, which indicates the type of the frame, that is initiate or domain segment and request or response. The command specifier is either termed as client command specifier (*ccs*) for the client device or server command specifier (*scs*) for the server device. For the initial request/response pair byte 0 also determines which of the three modes is used (see [20]). If transmission errors are detected either on the client or the server side, data transfer is aborted through the SDO abort frame. SDOs are used for configuration and parametrization, but also allow the transmission of a large quantity

of asynchronous data, consequently they are always assigned a lower priority than PDOs.

4.2.5 Predefined objects

These specific objects provide additional functionalities to the protocol. Their transmission is following the producer/consumer communication model. Particularly, the SYNC and the TIME object are always transmitted from a specific device (Producer), according to the OD specification, whereas the EMCY object can be transmitted by any device in the network (dynamical configuration). The Predefined objects are always assigned with a high priority, in order to be transmitted as soon as possible.

The SYNC object is used to enable synchronized operation. Yet, if the transmission is handled by the CAN protocol the derived delays due to non-preemption can result to a certain jitter. Thus, if it does not provide the required accuracy for the synchronization, CANopen enables the use of the TIME object, containing a reference clock time. Though implementing a different synchronization mechanism, this object is used for accuracy, measuring the difference between theoretical and the actual transmission time of the SYNC and transmit it through a subsequent PDO.

The EMCY object is used in internal error conditions in a device and transmitted as an interrupt, in order to notify other devices. However, no notification is present when the internal error is fixed and thus the other devices cannot know the change of condition. Consequently, its implementation is not considered mandatory in CANopen systems.

4.2.6 Network configuration

Considerable complexity in CANopen systems is found in the configuration and allocation of a frame identifier to each COB. As CANopen allows parametrization the allocation scheme can be configured according to specific manufacturer requirements, however sufficient attention must be given to the priority group of each object. Therefore, to reduce the complexity of CANopen system development, a default allocation scheme is provided for applications using CAN as the lower-layer communication protocol. This scheme is named Predefined Connection Set. As defined by this scheme, every object is assigned an identifier (COB-ID) according to Table 5, derived from its priority in the protocol. Nevertheless, each frame has its own identifier, since the COB-ID is augmented by the specific identifier of the node transmitting it. Every device can use up to four TPDOs, four RPDOs, one EMCY and one SDO. All the COB-IDs can be configured, except of the SDOs, if the particular device allows it.

Communication Object	COB-ID
NMT	0
SYNC	128
EMCY	129
TIME	256
TPDO1	385-511
RPDO1	513-639
TPDO2	641-767
RPDO2	769-895
TPDO3	897-1023
RPDO3	1025-1151
TPDO4	1153-1279
RPDO4	1281-1407
Tx-SDO	1408-1535
Rx-SDO	1536-1663

Table 5: Predefined Connection Set

4.2.7 Application development with CANopen

Apart from its use in automotive systems as a high-level protocol on top of CAN, CANopen can be used as an application layer protocol in industrial automation systems where it is integrated with Real-Time Ethernet technologies. The integration with many of those technologies is facilitated by the existence of a gateway [54] or a proxy [4]. Nevertheless, a possible constraint in this case are the additional latencies that often have a strong impact in the real-time performance. Therefore, a direct integration of CANopen as an application layer protocol along with its communication and device profiles is often preferred as an industrial solution. Two technologies facilitating this integration are Ethernet POWERLINK (EPL) (Section 4.2.8) and EtherCAT [43]. In particular, it can be integrated with many of Real-Time Ethernet technologies, however this usually implies the. Both technologies support fully the CANopen communication profile, in order to describe its services and mechanisms into a real-time Ethernet environment. Nevertheless, in many implementations EPL is preferred over EtherCAT, since it does not require any specific hardware modifications and is more suitable for the transmission of large amounts of data.

4.2.8 Ethernet Powerlink (EPL)

ETHERNET Powerlink (EPL) [49] is a commercial protocol for industrial automation systems based on the Fast Ethernet IEEE 802.3. One of protocol's major advantages is that it can operate with either the use of Ethernet switches or hubs, depending on the temporal constraints of the application. To overcome the effect of collisions occurring in standard Ethernet systems, EPL uses a TDMA technique (deployed in the data link layer), which is based on a mixed polling and time slicing mecha-

nism, called Slot Communication Network Management (SCNM) (Figure 17). This technique uses a special node, referred as Managing Node (MN), to grant the slave devices, referred as Controlled Nodes (CN's), access to the medium only when they are polled. The use of SCNM hampers the direct deployment of standard Ethernet devices in the network, as they would corrupt the access mechanism. To overcome this limitation dedicated gateway are connected to control the communication traffic of standard Ethernet devices. The supported topologies in EPL are the line and star topology.

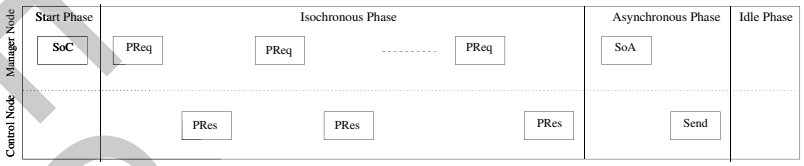

Fig. 17: EPL cycle

EPL supports periodic and event-based data exchange during a cyclic period of fixed duration. This period is divided in four phases, namely the starting, the isochronous, the asynchronous and the idle phase. The synchronized transition between phases is done through broadcast frames initiated by the MN device. More specifically, the reception of the Start of Cycle (*SoC*) frame by the slave devices ends the starting phase and accordingly begins the isochronous (cyclic) phase. During this phase the MN polls progressively every CN through a *PReq* unicast frame, in order to receive their data responses through the subsequent *PRes* frames. The *PRes* frames are also broadcasted, in order to facilitate data distribution amongst all the remaining nodes. Having polled all the CN devices in the EPL network, the MN broadcasts the Start of Asynchronous (*SoA*) frame, to indicate the beginning of the asynchronous period. This period allows a single asynchronous transaction (*Send* in Figure 17) to be performed. This transaction might be an asynchronous EPL data frame (*ASnd* frame), detection of active stations (*IdentRequest* frame), or even a standard Ethernet data frame. All the asynchronous transactions are enqueued in the MN, in order to be transmitted according to their priority. As the asynchronous period is used for the exchange of large frames, the EPL cycle includes the idle phase to ensure that the ongoing transaction has ended.

4.2.9 EPL frame format

EPL frames (Figure 18) are encapsulated and transmitted in the Data field of IEEE 802.3 standard Ethernet frames. Therefore, their main difference with the legacy Ethernet frames is the Ethernet Type field of the Ethernet frame, which is set to the hexadecimal value 88ABh. An EPL frame consists of five fields: the Message Type,

specifying the type of EPL frame (as defined above), the EPL source and destination addresses, the EPL data to be exchanged and an optional padding. The Message Type field can contain one of the values present in Table 6.

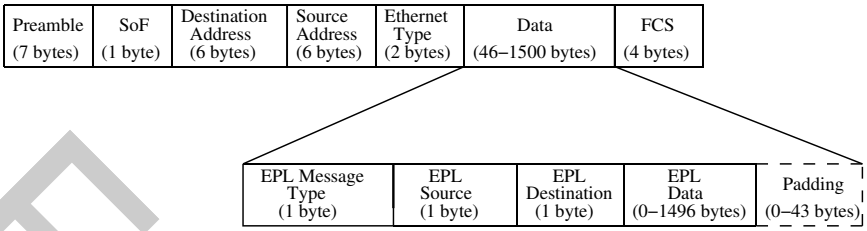

Fig. 18: EPL frame format

Message Type	EPL frame type
01	Start of Cyclic (SoC)
03	Poll Request (PReq)
04	Poll Response (PRes)
05	Start of Asynchronous (SoA)
06	Send (ASnd or IdentRequest)

Table 6: Message Type field of EPL frame

The EPL source and destination addresses for each frame are specified according to the Table 7. Specifically, we can denote that for each EPL node there is a unique address, except from the MN which is always equal to 240. Likewise, the CN's have an id in the range 1-239 and 253 is used by a node to address itself. Finally, as the minimum Ethernet II frame size is equal to 64 bytes, extra padding data is added to the packet. The size of the data padding in an EPL frame can be up to 43 bytes.

4.2.10 Application layer profile

EPL is fully integrated with the CANopen protocol as well as its communication and device profiles as presented in Section 4.2.1. This integration facilitates the description of its services and mechanisms into a real-time Ethernet environment.

As a result of the integration, CANopen's objects are encapsulated into lower-layer EPL frames. Initially, during the isochronous phase of the EPL cycle (Figure 17), data relevant to the application are stored and exchanged through Process Data Objects (PDOs). To this regard, the MN sends a TPDO to each CN via a PReq frame which, in turn, stores the data in one or more RPDOs and responds with a TPDO

Address	EPL node
0	Invalid
1-239	Controlled Node (CN)
240	Managing Node (MN)
241-252	Reserved
253	Self diagnostic identifier
254	EPL to legacy Ethernet router
255	Broadcast identifier

Table 7: EPL node addressing

encapsulated in a PRes frame. Moreover, in the asynchronous phase configuration data are exchanged through Service Data Objects (SDOs), respectively encapsulated in ASnd frames.

EPL also uses the Object Dictionary (OD) to store all the network-accessible data. It may also contain a maximum of 65536 entries as well distinguished in the communication, manufacturer and device specific categories. The OD entries are described only by the XDD and XDC file formats, similar to the one presented in Figure 15.

Even though the CANopen communication profile is fully integrated in EPL, there are also minor differences between them as for with the SDO channels, which is are defined and configured during initialization in CANopen, but instead EPL allows a dynamical configuration of these channels. Another difference lies on the transmission of unconfirmed segment frames during the segmented data transfer in EPL, whereas as mentioned in Section 4.2.1 CANopen segments are always confirmed. A method for confirming the transmission when large amounts of configuration data need to exchanged is through the use of multiple expedited SDO transfers.

4.2.11 Application development with EPL

When developing applications in EPL the most frequent challenges (by priority level) that may arise are:

1. **Separation of functionalities between the EPL nodes.** The developer should be able clarify and implement a different behavior for each EPL node, according to the type of EPL application. As an example in a sense-compute-control application the MN node is not only used for polling, but the CN's may often require dedicated data from it, in order to perform actuations. This is handled in the EPL cycle by supporting transmission capabilities to the MN node using proper configuration. Therefore, the developer should be able clarify and implement a different behavior for each EPL node.
2. **Mapping of application-specific functionality to the Object Dictionary entries.** Once a clear functionality separation is defined, the developer should assigns

specific entries to the Object Dictionary for handling the network configuration as well as the exchange of time critical or asynchronous data in the application. This task should be done in respect to the CANopen profile and thus requires high expertise, in order to define the correct data encoding and object linking and may be time consuming if the application's behavior is complex.

3. **Selection of the EPL configuration parameters.** EPL applications are characterized by strict timing constraints. Therefore the selection of parameters, such as the cycle duration, the timeout for acquiring the polling responses, the tolerance timeout in the CN's for receiving the SoC frame and the maximum transmitted data during the asynchronous phase, determines to a large extent the EPL application functionality. The selection of these parameters also depends on the characteristics of resource-constrained devices (e.g. computational platforms), which are chosen in the underlying hardware architecture.

From the aforementioned challenges we can reason that the correct configuration of the MN and the CN devices is of vital importance in EPL application development. To this end, an open source (BSD Licence) tool for the development of applications with Ethernet Powerlink (EPL) was defined, named openPOWERLINK [6]. openPOWERLINK is an open source (BSD Licence) Real-Time Ethernet stack provided by SYSTEC electronic [16]. openPOWERLINK is developed using a layered approach, which segments the system in a hierarchical way, namely the user and the kernel part. The former implements the application layer of the EPL protocol and provides an API for the development of EPL applications. It contains an implementation for the OD, as well as the PDO, SDO, Error Handler and Event Handling modules. The latter implements the Data Link Layer (DLL) of the EPL protocol and the necessary drivers to communicate with the hardware. It also contains an Event Handling module as well as implementations for an Ethernet and a time-critical driver (for the time slicing mechanism). The Event Handling module is responsible for delivering events, which are related to object dictionary accesses, completion of SDO transfers, configuration and stack errors etc. The two parts interact with each other by message passing through the Communication Abstraction Layer (CAL). All the processes defined above the CAL have a high-priority in the stack, whereas the ones below have a low-priority. The overall architecture of the openPOWERLINK stack is illustrated in Figure 19. In order to allow NMT functionalities related to the CANopen protocol (Section 4.2.2) openPOWERLINK supports an additional NMT module to manage the NMT state machine. The Managing Node can use this module to set its or the Controlled Nodes' state machines into four states, namely PreOperational1, PreOperational2, ReadyToOperate and Operational. In the PreOperational1 state all the modules are stopped, the PreOperational2 allows the functionality all the modules except from the PDO and the ReadyToOperate is a transitional state where the PDO module before moving to the Operational state.

openPOWERLINK handles communication between different layers of the stack using dedicated methods, such as *variable linking*. This method defines specific API variables, called *process variables*, which are accordingly linked with entries of the

[16] http://www.systec-electronic.com

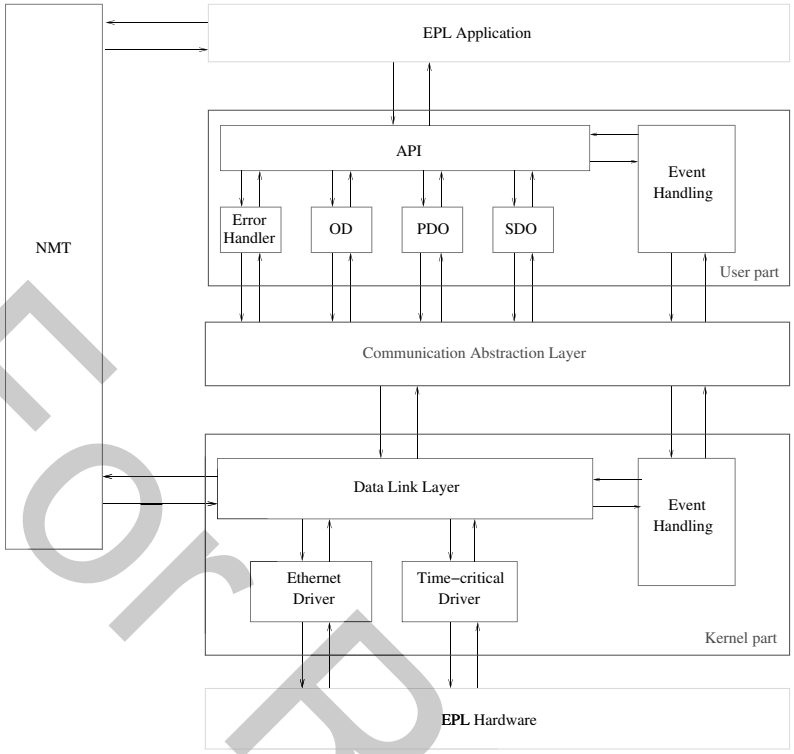

Fig. 19: openPOWERLINK stack architecture

OD. The process variables are also used to realize communication in the isochronous phase of the EPL cycle through the PRes frames. Two main types of process variables are used in the stack, namely input and output process variables. The former handle the processing as well as manipulation of sensor/actuator data (i.e. ATD conversion, encoding/decoding, data display) and are also initiating data transmission. On the other hand the latter are useful for data reception.

Further techniques in the openPOWERLINK stack is the assignment of higher priority to data handling during the EPL cycle than event handling. This ensures its real-time behavior and is accomplished with the use of threads. Furthermore, it supports real-time communication between modules of one or several layers is the presence of *asynchronous callbacks*. Callbacks are functions that are passed as an argument to another function and are commonly using in event-driven programming. In this way a callback is used to subscribe to an event and accordingly invoked when the event happens. As an example, in openPOWERLINK callbacks are used in the EPL Application for data transmission in which case a callback has to be defined for the communication with the user as well as the kernel part. The most commonly used callbacks in openPOWERLINK define communication between the Communication Abstraction Layer (CAL) and Data Link Layer for the kernel part as well as between

the EPL Application and the API layer for the user part. The former is used to update the values of the process variables during the EPL cycle and the latter to provide event-based notifications from the user part modules (i.e Event Handler, OD, PDO, SDO) to the EPL Application.

Until now we have presented techniques which are used in openPOWERLINK to successfully address and handle the aforementioned challenges. The described techniques can be used for the development of functional applications in openPOWERLINK, which is sequential and relies on the following steps.

1. The MN should detect and access the connected CNs in the EPL network through an Ident Request
2. The Object Dictionary entries in the CNs are initialized by dedicated SDO frames in the asynchronous phase of the EPL cycle
3. The process variables of the EPL Application layer should be linked with entries of the OD module for each node (MN or CN). Once linked, a modification of a process variable will automatically signal the API layer to update the dedicated entry in the node's OD.
4. Implementation of the callback between the Communication Abstraction Layer (CAL) and the Data Link Layer for the kernel part.
5. Implementation of the callback between the EPL Application and the API layer for the user part.

Application development for a Controlled Node

The CN should be able to reply in certain time frames whenever polled by the MN device. It is configured according to the following steps:

1. Reply to the MN's Ident Request
2. Link the API process variables to the OD
3. Store the API process variables in the OD
4. Implement the callback function called by the event handling module in the kernel part
5. Implement the callback function called by the API layer in the user part

4.2.12 EtherCAT

EtherCAT[17] (Ethernet for Control Automation Technology) is developed by Beckhoff Automation Inc. as an Ethernet-based Fieldbus system. The main objective was using Ethernet in automation applications with short response time ($\leq$ 100 μs) and low jitter for accurate synchronization ($\leq$ 1 μs)). To sum up, the EtherCAT protocol is an optimized Ethernet protocol devised for short cyclic process data.

[17] More information available at:
https://www.ethercat.org/en/technology.html

4.3 Future directions: Interaction with the IoT ecosystem

4.3.1 Wireless and cellular infrastructure

Industrial devices (e.g. PLCs) can communicate with the Cloud to perform data processing and analysis as well as allow remote maintenance and troubleshooting from operation engineers. Data exchange is can be performed by integrating the industrial devices with solutions that belong in 3 main categories:

- *5G radio access technologies*: This technology provides wide area, broadband access. The 5G technology is currently in the process of conceptual development and standardization by the World Radiocommunication Conference (WRC). The 5G technology is expected to have a specific V2X aspect of the 5G technology in a practical scale after 2020. However, in this document we are leveraging the limited standardization to illustrate conceptually its main scope and architectural view.
- *Pre-5G radio access technologies*: Multiple cellular technologies were identified by the ETSI 3rd Generation Partnership Project (3GPP), *LoRa* Alliance and other organizations, such as Narrowband IoT [46] , Long Term Evolution for Machines (LTE-M) [44], LoRa are [38] considered. Even though these technologies are already used in V2P/P2V, the main challenge when adopting them in other V2X communication types are reliability and safety, which are currently not addressed in the scope of Low-Power Wide Area Networks (LPWAN).
- *Non-cellular technologies providing wireless access*: IEEE has defined different standards for wireless communication, such as 802.11ac and 802.11p, however only 802.11p is flexible in terms of throughput and offers higher reliability, even though its maximal throughout is more limited than 802.11ac (from 3 to 27 Mbps raw data rate). The reason behind this is that 802.11p was designed particularly for for safety-related Vehicular Ad-hoc NETworks (VANET), including the V2V and V2I/I2V concepts. IEEE 802.11p technology is currently fully specified and already deployed in different locations.

The following paragraphs start with a description of the scenarios supported by 802.11p communication and cellular communication. This is followed by a description on both the 802.11p and 5G technologies. In the scope of this section we focus on these two technologies, because, to the best of our knowledge, they are considered as the leading candidates for V2X communication.

4.3.2 Connected cars

Industrial IoT technologies allow vehicles to evolve by the integration of technologies that allow them to take decisions autonomously without any human intervention. This allows vehicles to communicate with many external entities in context that is now referenced as Vehicle to Everything (V2X). V2X enables the following scenarios in vehicles:

- Vehicle-2-Vehicle (V2V)
- Vehicle-2-Infrastructure/Infrastructure-2-Vehicle (V2I,I2V)
- Vehicle-2-Pedestrian (V2P) / Pedestrian-2-Vehicle (P2V)
- Vehicle-2-Network (V2N) / Network-2-Vehicle (N2V), 5) Infrastructure-2-Network (I2N) / Network-2-Infrastructure (N2I).

These types along with their interactions are demonstrated in Figure 20.

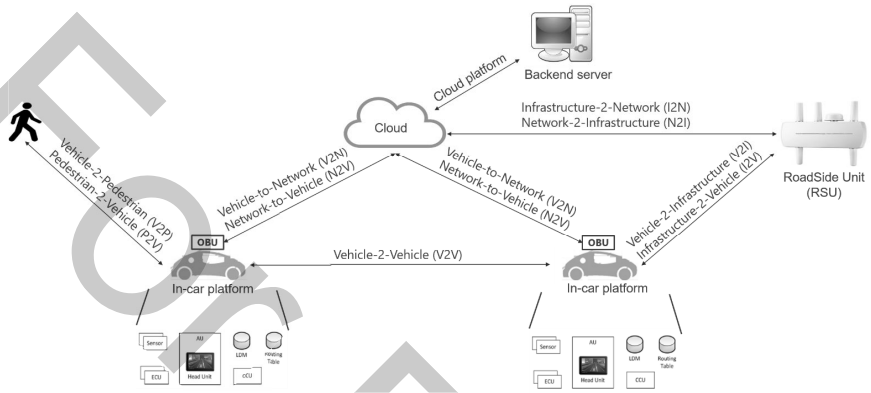

Fig. 20: V2X communication types

The ultimate goal of the integration would be the fleet management by each industrial manufacturer, that would allow remote diagnostics and monitoring for each factory-built vehicle. Specific use-cases for fleet-management are:

- Fault prediction based on analytics
- Remote maintenance
- Energy management

4.3.3 Smart grid (e.g. DNP3, DLMS/COSEM protocols)

Industrial IoT technologies bring a revolution also to the traditional power grid. Specifically, the traditional power grid consisted of isolated small systems and was limited to small geographical areas, which led to local power generation, local limited transmission capabilities and local power distribution. Instead by integrating IoT technologies the power grid transitions towards the smart grid, where all the systems are interconnected. To understand more on the interconnection of smart grid systems we illustrate in Figure 21, the four main systems that they are composed of as well as the assets (i.e. devices) that handle data exchange in them [40]. A detailed explanation of the main systems also follows:

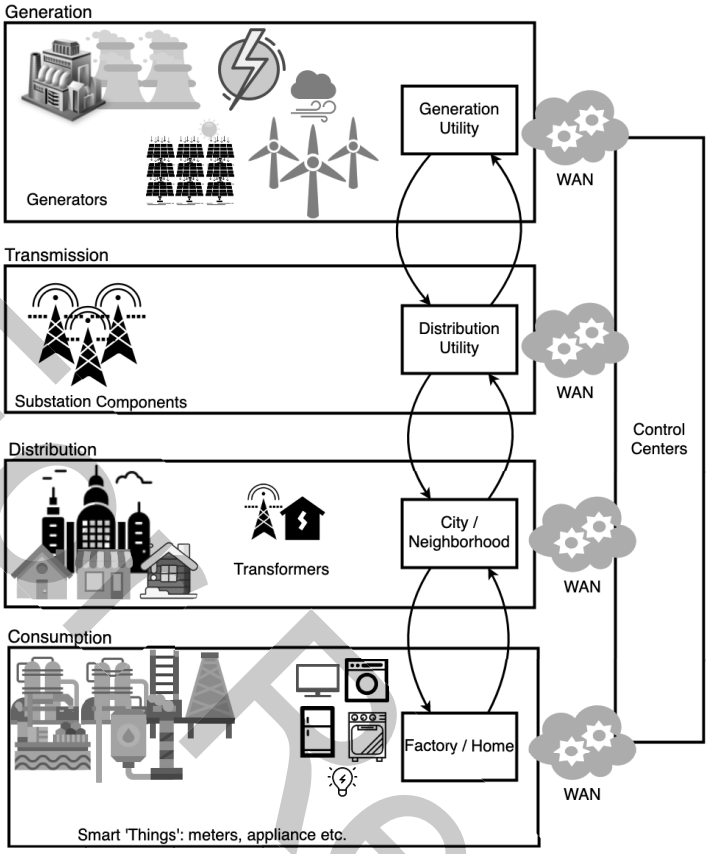

Fig. 21: Smart grid architecture based on NIST Smart Grid Conceptual Model.

- Generation: The utility company has dedicated assets, called generators, to provide substations with enough energy, which they can distribute to the utility substations.
- Transmission: The utility substation of the energy distribution company is responsible for distributing the electricity to entire cities or neighbourhoods. In this system we can also find switching, protection and control equipment, and circuit breakers in order to interrupt any short circuits or overload currents that may occur on the network.
- Distribution: It is handling the dispatch of energy through the Utility Access Points and associated assets, called transformers, responsible for converting high voltage power used in power lines to lower voltage power that can be used safely in residential homes and businesses. This system also includes data concentrators, that are responsible for calculating the difference between the actual energy consumption in each individual house and compares to an estimated consumption

that is associated with historical consumption data for each house as well as with the climate conditions in the house area.
- Consumption: This system concerns the usage of electricity in the consumer houses or factory buildings that employ smart appliances and smart meters in order to manage and optimize the energy consumption in the system. Smart meters are an important source of distributed intelligence for the smart grid and hence are often referred to in literature as Advanced Metering Infrastructure (AMI).

If we focus now on the communication protocols that enable the transition to the smart grid we observer three main protocols that are used for data exchange: DLMS/COSEM and DNP3.

DLMS/COSEM

DLMS/COSEM is a widely used communication protocol for metering devices. DLMS stands for "Device Language Message specification" and describes a concept for modeling of communication entities; COSEM stands for "COmpanion Specification for Energy Metering" and sets the rules for data exchange with energy meters. The application fields for DLMS/COSEM are in electricity metering or gas metering within the commercial & industrial environment. Also other applications are feasible, for example heat or water.

DLMS/COSEM represents a client-server architecture. In common operation mode the client is represented by the Head-End System that connects to the metering device, which represents the server. Furthermore, push mechanisms are also available for sending information for the server directly to the client. For example this needed when sending critical alarms from a metering device to the Supervisory Control And Data Acquisition (SCADA) System.

DLMS/COSEM defines the following for smart meter communication:

- An object model, to view the functionality of the meter, as it is seen at its interface(s)
- An identification system for all metering data
- A messaging method to communicate with the model and to turn the data to a series of bytes
- A transporting method to carry the information between the metering equipment and the data collection system

DLMS/COSEM specification covers five ISO/OSI communication layers that are depicted in Figure 22. Specifically, DLMS/COSEM supports both HDLC [21] and IP [22] transport layers. The smart meter software provided supported HDLC, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) transports.

The P0 optical port (Figure 23) on some smart meters support IEC-62056-21 [23] mode E, or DLMS / COSEM over HDLC. The protocol for requesting the meter to enter mode E is the same as is described in the standard but in practice, timing was an big issue in setting up the physical connection.

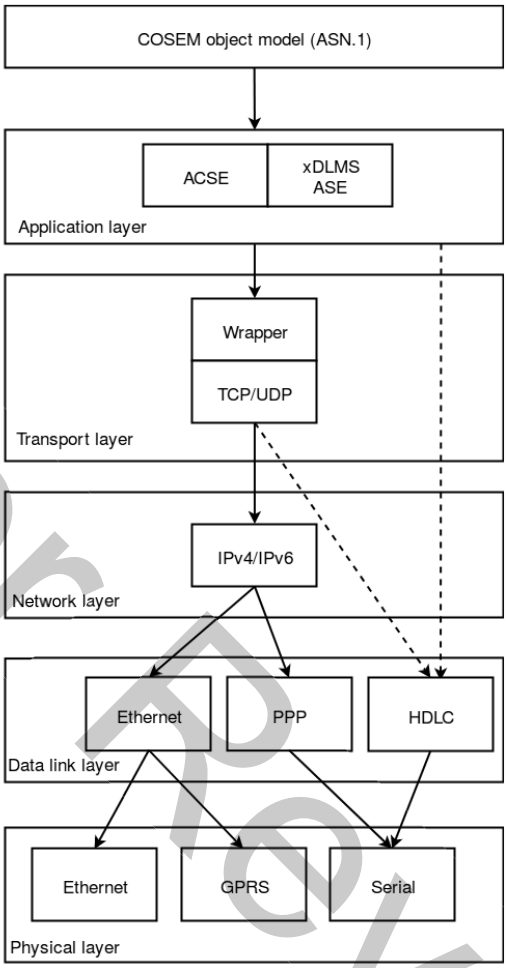

Fig. 22: DLMS/COSEM protocol stack

Security in DLMS/COSEM is specified through a data protection security layer that provides encryption and authentication mechanisms. Encryption is based on Elliptic curve digital signature (ECDSA) or Elliptic curve Diffie-Hellman (ECDH) key agreement [26]. The authentication mechanism includes three procedures: 1) no security, where no identification takes place, 2) Low Level Security (LLS), where the DLMS server identifies the DLMS client by password and 3) High Level Security (HLS) authentication, where both DLMS server and client and mutually identified by challenge exchange and verification of the challenge result Figure 24 shows the different authentication mechanisms between a DLMS client and a server.

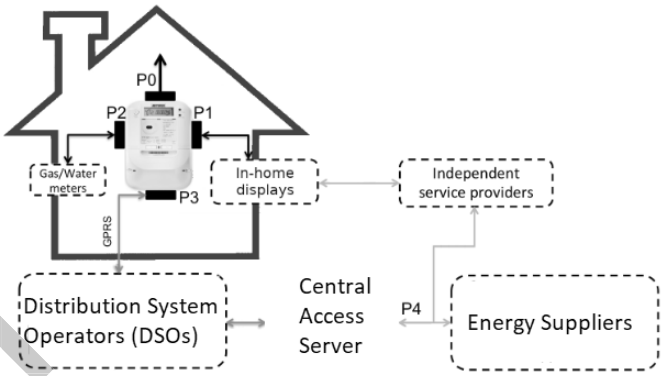

Fig. 23: Smart meter ports that generate data

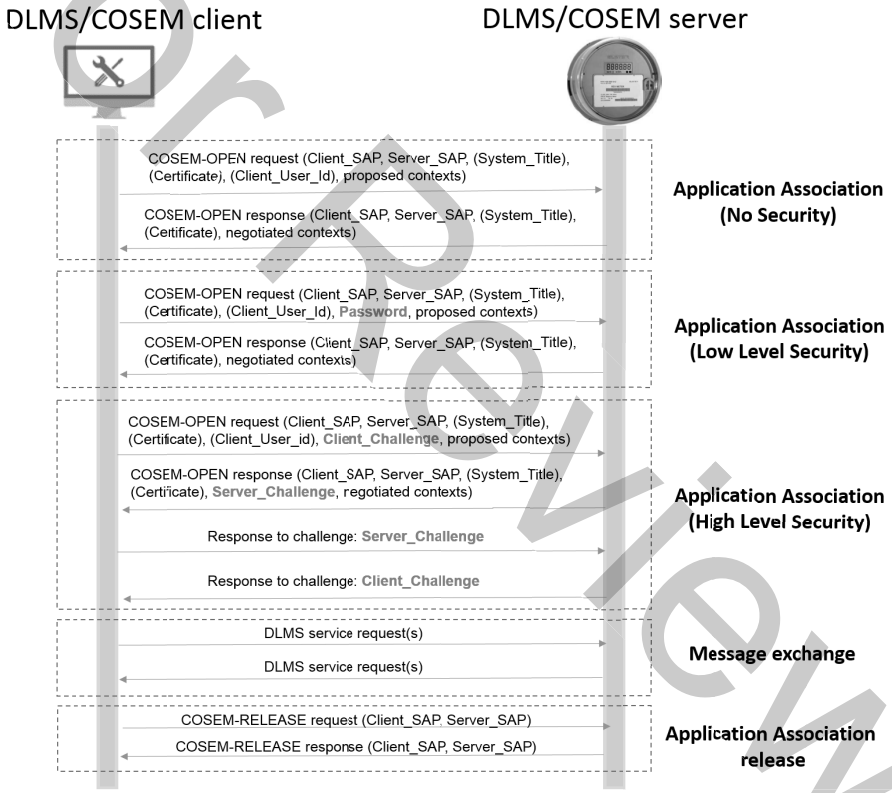

Fig. 24: Authentication mechanisms between Workstation (DLMS client) and smart meter (DLMS server)

DNP3

Distributed Network Protocol-3 (DNP3) is published by IEEE in 2010 and is a standard based on the IEC 61850 standard, which defines a set of communication protocols used in process automation systems, especially utility distribution, such as electricity and water. The protocol is developed to assess communication in between various types of data acquisition and control equipment, such as SCADA(s), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs) [16].

4.3.4 Building automation system (e.g. ZigBee, BACnet)

Building Automation (BA) is designed to automate and control buildings' air conditioning, elevator, heating, lighting, and other smart devices via Building Automation System (BAS) to improve the comfort and reduce the overall energy cost. BA is a good example of distributed control systems since there could be multiple gas/smoke detectors, light, security, temperature, water sensor in the building. Fiber, Ethernet or wireless communication is used to provide physical connectivity between the devices. Protocols like BACnet, ZigBee and various other widely used in BA. BACnet (Building Automation and Control Networks) leverages the ASHRAE, ANSI, and ISO 16484-5 standard protocols.

ZigBee is a short-range (10-100 meters) low power (low energy consumption) wireless communication protocol based on IEEE 802.15.4 standard.

5 Conclusions

In particular, following are the main contributions of this chapter:

1. New advancements in the field of industrial networks, especially IIoT are provided.
2. Security related implications and projections are provided.
3. Most of the prominent networking and communication technologies are provided in a comparative way.

Authors of this chapter foresee that future industrial networks, especially their IDSs, will take advantage of data streaming concept in their algorithms, due to several reasons: 1) To provide a rapid response by removing the necessity of the round-trip messages in between the edge of the network and the server, 2) to decrease the amount of traffic created hence causing less bottlenecks in the network, and finally, 3) to reduce the overall response time for the queries by processing the data closer to the source.

As a conclusion, future industrial and IIoT networks will benefit from various emerging advanced technologies such as LoRa, Nb-IoT, 5G, SDN, SDR, Fog/Edge, etc., to be able to implement the needs of the Industry 4.0 such as remote observing, monitoring, and commanding features along with the Digital Twin.

References

1. Industrial IoT: Rise of Digital Twin in Manufacturing Sector. https://www.biz4intellia.com/blog/rise-of-digital-twin-in-manufacturing-sector/. Accessed: 2020-1-26
2. Measuring the Information Society Report. https://www.itu.int/en/ITU-D/Statistics/Documents/bigdata/MIS2015_Chapter5.pdf. Accessed: 2020-1-26
3. Alley, G.: What Is Data Streaming? https://dzone.com/articles/what-is-data-streaming. Accessed: 2019-07-12
4. Axel Pöschmann Lutz Rauchhaupt, T.W.: Integration of CAN-based Networks into the PROFInet Environment. In: 9th International CAN Conference, Munich, Germany (2003)
5. Aydogan, E., Yilmaz, S., Sen, S., Butun, I., Forsström, S., Gidlund, M.: A central intrusion detection system for rpl-based industrial internet of things. In: 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1–5. IEEE (2019)
6. Baumgartner, J., Schoenegger, S.: Powerlink and real-time linux: A perfect match for highest performance in real applications. In: Twelfth Real-Time Linux Workshop, Nairobi, Kenya (2010)
7. Bizanis, N., Kuipers, F.A.: Sdn and virtualization solutions for the internet of things: A survey. IEEE Access **4**, 5591–5606 (2016)
8. Butun, I.: Prevention and detection of intrusions in wireless sensor networks. University of South Florida, Ph.D. thesis (2013)
9. Butun, I.: Privacy and trust relations in internet of things from the user point of view. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1–5. IEEE (2017)
10. Butun, I., Morgera, S.D., Sankar, R.: A survey of intrusion detection systems in wireless sensor networks. IEEE communications surveys & tutorials **16**(1), 266–282 (2013)
11. Butun, I., Österberg, P.: Detecting intrusions in cyber-physical systems of smart cities: Challenges and directions. In: Secure Cyber-Physical Systems for Smart Cities, pp. 74–102. IGI Global (2019)
12. Butun, I., Österberg, P., Song, H.: Security of the internet of things: Vulnerabilities, attacks and countermeasures. IEEE Communications Surveys & Tutorials (2019)
13. Butun, I., Pereira, N., Gidlund, M.: Analysis of lorawan v1. 1 security. In: Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects, p. 5. ACM (2018)
14. Butun, I., Pereira, N., Gidlund, M.: Security risk analysis of lorawan and future directions. Future Internet **11**(1), 3 (2019)
15. Butun, I., Sari, A., Österberg, P.: Security implications of fog computing on the internet of things. In: 2019 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6. IEEE (2019)
16. Butun Ismail, L.A., dos Santos, D.: Security and privacy in smart grids: Challenges, current solutions and future opportunities. In: in Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP), pp. 1–6. Springer (2020)
17. CAN in Automation: Application Note 802 (2005)
18. CAN in Automation: Electronic data sheet specification for CANopen, Draft Standard 306 (2005)
19. CAN in Automation: CANopen device description, Draft Standard 311 (2007)
20. CAN in Automation: Application layer and communication profile, Draft Standard 301 (2011)
21. Commission, I.E., et al.: Electricity metering - data exchange for meter reading, tariff and load control. IEC-62056 Part 46 **21**, 31–51 (2002)
22. Commission, I.E., et al.: Electricity metering - data exchange for meter reading, tariff and load control. IEC-62056 Part 47 **21**, 31–51 (2002)
23. Commission, I.E., et al.: Electricity metering data exchange - the dlms/cosem suite-part 5-3: Dlms/cosem application layer. Tech. rep., IEC 62056-5-3: 2016, ed (2016)

24. Decotignie, J.D.: Ethernet-based real-time and industrial communications. Proceedings of the IEEE **93**(6), 1102–1117 (2005)
25. Derhamy, H., Eliasson, J., Delsing, J.: Iot interoperability-on-demand and low latency transparent multiprotocol translator. IEEE Internet of Things Journal **4**(5), 1754–1763 (2017)
26. DLMS, U.: Dlms/cosem architecture and protocols. Green Book (2019)
27. Eldefrawy, M., Butun, I., Pereira, N., Gidlund, M.: Formal security analysis of lorawan. Computer Networks **148**, 328–339 (2019)
28. Farris, I., Taleb, T., Khettab, Y., Song, J.: A survey on emerging sdn and nfv security mechanisms for iot systems. IEEE Communications Surveys & Tutorials **21**(1), 812–837 (2018)
29. Farwell, J., Rohozinski, R.: Stuxnet and the future of cyber war. Survival **53**, 23–40 (2011). DOI 10.1080/00396338.2011.555586
30. Ferrari, P., Sisinni, E., Brandão, D., Rocha, M.: Evaluation of communication latency in industrial iot applications. In: 2017 IEEE International Workshop on Measurement and Networking (M&N), pp. 1–6. IEEE (2017)
31. Grzywaczewski, A.: Training ai for self-driving vehicles: the challenge of scale (2019). URL https://devblogs.nvidia.com/training-self-driving-vehicles-challenge-scale/
32. Halpern, M.: Iran flexes its power by transporting turkey to the stone age. Observer (2015). URL https://observer.com/2015/04/iran-flexes-its-power-by-transporting-turkey-to-the-stone-ages/
33. Hatzivasilis, G., Askoxylakis, I., Alexandris, G., Anicic, D., Bröring, A., Kulkarni, V., Fysarakis, K., Spanoudakis, G.: The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0. In: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–7. IEEE (2018)
34. Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., Eschert, T.: Industrial internet of things and cyber manufacturing systems. In: Industrial internet of things, pp. 3–19. Springer (2017)
35. Koc, A.T., Jha, S.C., Gupta, M., Vannithamby, R.: Extended discontinuous reception (drx) cycle length in wireless communication networks. U.S. Patent Application No **14** (2013)
36. Kocakulak, M., Butun, I.: An overview of wireless sensor networks towards internet of things. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1–6. IEEE (2017)
37. Lekidis, A.: Design flow for the rigorous development of networked embedded systems. Ph.D. thesis (2015)
38. LoRa Alliance: A technical overview of lora and lorawan (2015)
39. Mitola, J.: Software radios: Survey, critical evaluation and future directions. IEEE Aerospace and Electronic Systems Magazine **8**(4), 25–36 (1993)
40. Mo, Y., Kim, T.H.J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., Sinopoli, B.: Cyber–physical security of a smart grid infrastructure. Proceedings of the IEEE **100**(1), 195–209 (2011)
41. Najdataei, H.: Parallel data streaming analytics in the context of internet of things. Licentiate Thesis, Chalmers University of Technology (2019)
42. Pfeiffer, O., Ayre, A., Keydel, C.: Embedded networking with CAN and CANopen. Copperhill Media (2008)
43. Prytz, G.: A performance analysis of ethercat and profinet irt. In: Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, pp. 408–415. IEEE (2008)
44. Ratasuk, R., Mangalvedhe, N., Ghosh, A., Vejlgaard, B.: Narrowband LTE-M system for M2M communication. in Vehicular Technology Conference (VTC Fall), IEEE
45. Ratasuk, R., Mangalvedhe, N., Zhang, Y., Robert, M., Koskinen, J.P.: Overview of narrowband iot in lte rel-13. In: IEEE conference on Standards for Communications and Networking (CSCN) (2016)
46. Ratasuk, R., Mangalvedhe, N., Zhang, Y., Robert, M., Koskinen, J.P.: Overview of narrowband iot in lte rel-13. In: IEEE conference on Standards for Communications and Networking (CSCN) (2016)

47. Sagiroglu, S., Terzi, R., Canbay, Y., Colak, I.: Big data issues in smart grid systems. In: 2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA), pp. 1007–1012. IEEE (2016)
48. Std., I.E.C.: Iec 61784: Digital data communications for measurement and control - part 1: Industrial communication networks - profiles - part 1: Fieldbus profiles (August 2014)
49. Std., I.E.C.: Iec 61784: Digital data communications for measurement and control - part 2: Additional profiles for iso/iec8802-3 based communication networks in real-time applications (July 2014)
50. Steer, M.: Will There Be A Digital Twin For Everything And Everyone? www.digitalistmag.com. Accessed: 2018-10-08
51. Steiner, W.: Ttethernet specification. TTTech Computertechnik AG, Nov (2008)
52. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology **94**(9-12), 3563–3576 (2018)
53. Tello-Oquendo, L., Akyildiz, I.F., Lin, S.C., Pla, V.: Sdn-based architecture for providing reliable internet of things connectivity in 5g systems. In: 2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), pp. 1–8. IEEE (2018)
54. Zeltwanger, H.: Gateway profiles connecting CANopen and Ethernet. In: 10th International CAN Conference, Rome, Italy (2005)
55. Zhang, Y., Huang, T., Bompard, E.F.: Big data analytics in smart grids: a review. Energy Informatics **1**(1), 8 (2018)