Iceland Liechtenstein Norway grants

"Working together for a green, competitive and inclusive Europe"

Project: *Digitalisation of water sector and water education* - **DIGIWATRO**,
Contract: 20-COP-0050

Intellectual Output 2: *Promoting life-long learning in the water sector using hybrid education*

Disclaimer

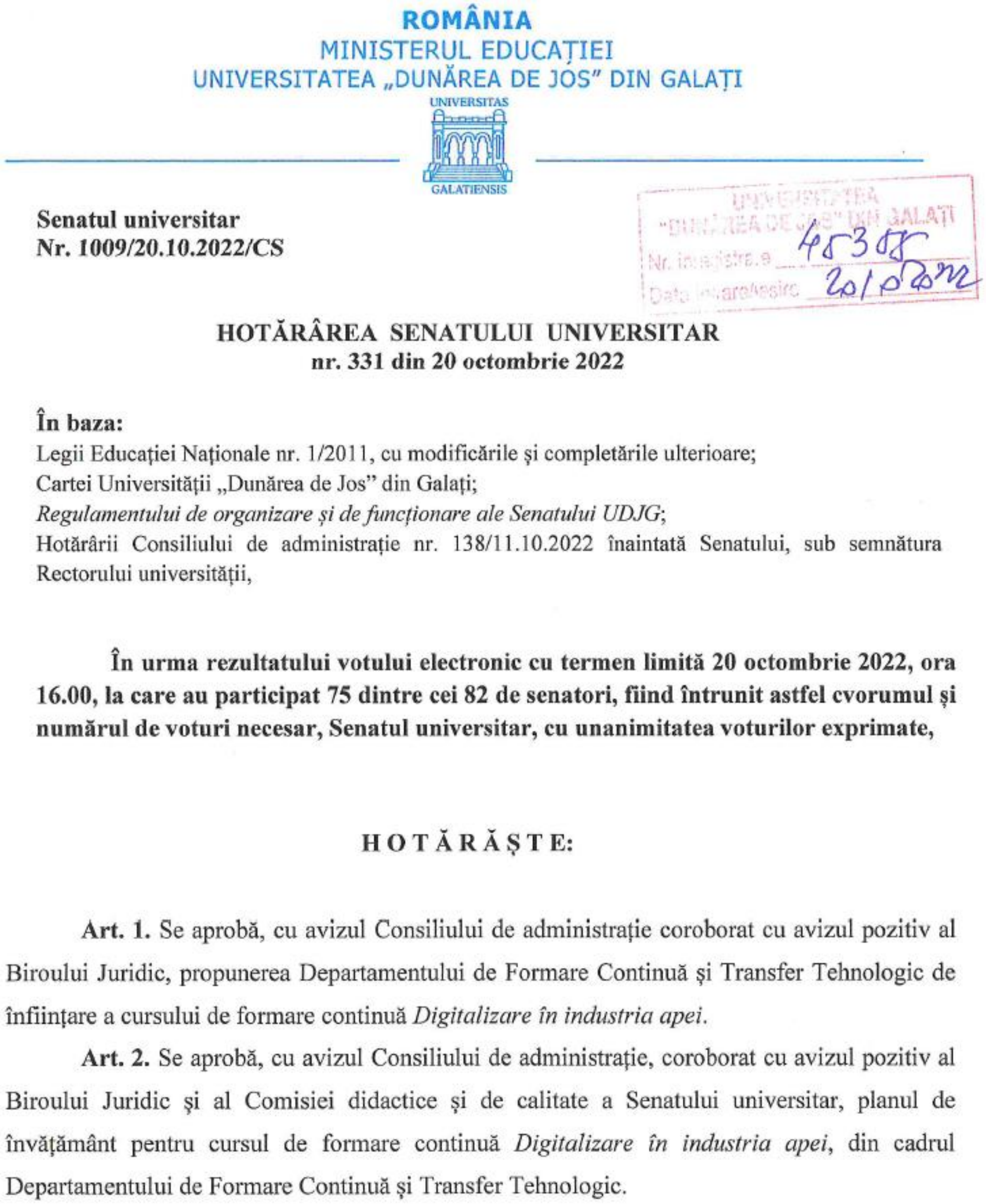
This document was realized with the EEA Financial Mechanism 2014-2021 financial support. Its content (text, photos, videos) does not reflect the official opinion of the Programme Operator, the National Contact Point and the Financial Mechanism Office. Responsibility for the information and views expressed therein lies entirely with the authors.

UNIVERSITAS GALATIENSIS

Norwegian University of Life Sciences

Introduction

Within this Intellectual Output, a continuous education program entitled "Digitalization in the water industry" was designed and carried out. The program was carried out within the Department of Continuous Education and Technology Transfer of "Dunarea de Jos" University of Galati, being approved by the University Senate Decision no. 331 of 20.10.2022 (print screen from the University Senate decision is given below).



ROMÂNIA
MINISTERUL EDUCAȚIEI
UNIVERSITATEA „DUNĂREA DE JOS" DIN GALAȚI

Senatul universitar
Nr. 1009/20.10.2022/CS

HOTĂRÂREA SENATULUI UNIVERSITAR
nr. 331 din 20 octombrie 2022

În baza:
Legii Educației Naționale nr. 1/2011, cu modificările și completările ulterioare;
Cartei Universității „Dunărea de Jos" din Galați;
Regulamentului de organizare și de funcționare ale Senatului UDJG;
Hotărârii Consiliului de administrație nr. 138/11.10.2022 înaintată Senatului, sub semnătura Rectorului universității,

În urma rezultatului votului electronic cu termen limită 20 octombrie 2022, ora 16.00, la care au participat 75 dintre cei 82 de senatori, fiind întrunit astfel cvorumul și numărul de voturi necesar, Senatul universitar, cu unanimitatea voturilor exprimate,

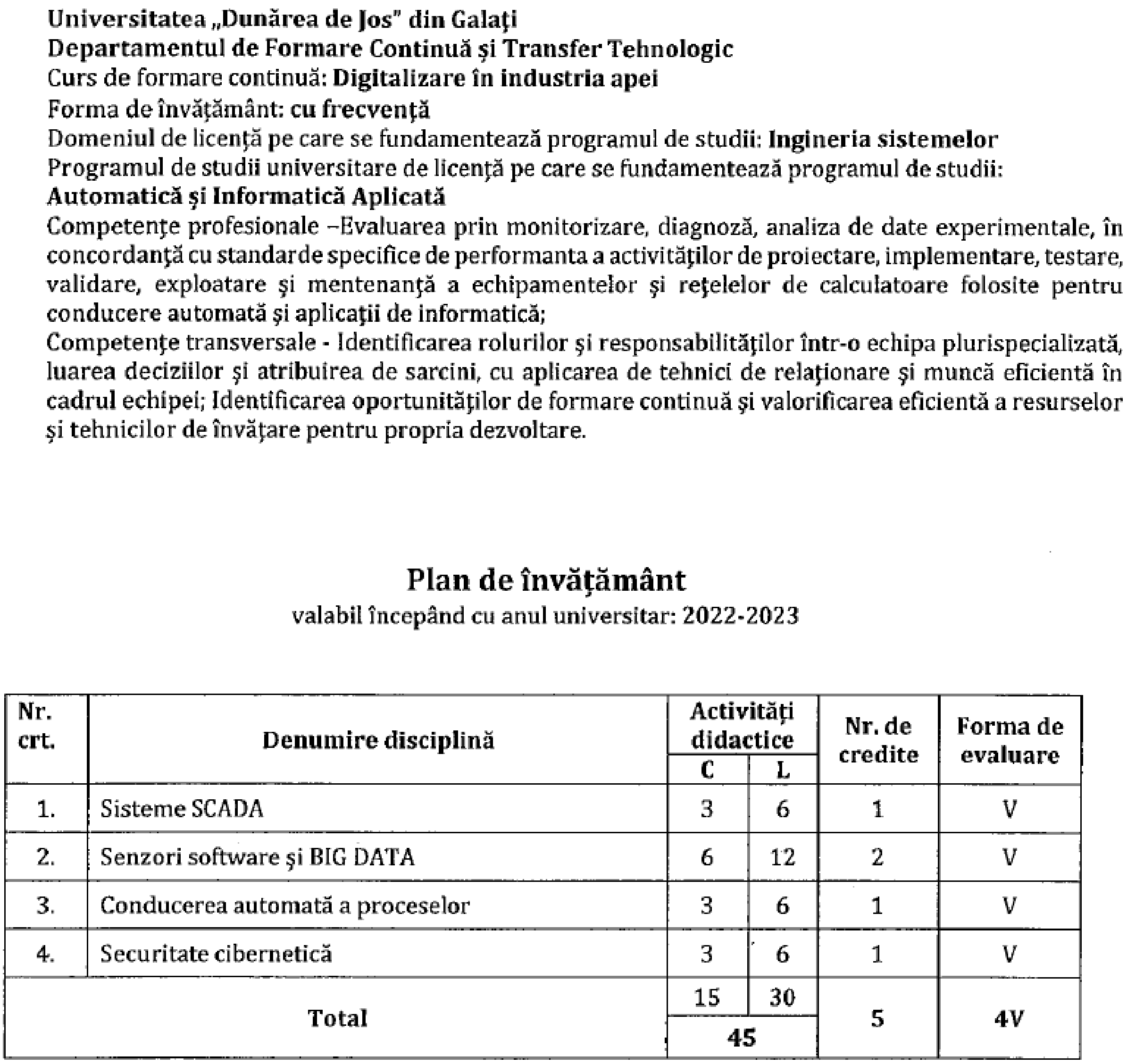
HOTĂRĂȘTE:

Art. 1. Se aprobă, cu avizul Consiliului de administrație coroborat cu avizul pozitiv al Biroului Juridic, propunerea Departamentului de Formare Continuă și Transfer Tehnologic de înființare a cursului de formare continuă *Digitalizare în industria apei*.

Art. 2. Se aprobă, cu avizul Consiliului de administrație, coroborat cu avizul pozitiv al Biroului Juridic și al Comisiei didactice și de calitate a Senatului universitar, planul de învățământ pentru cursul de formare continuă *Digitalizare în industria apei*, din cadrul Departamentului de Formare Continuă și Transfer Tehnologic.

The program contains 4 disciplines with a total number of 45 hours and 5 credits, 15 hours being allocated to courses and 30 hours being allocated to laboratory-type applicative activities. The included disciplines include updated content, in corelation with the current trend in digitalization, in an effort undertaken by the project to improve the knowledge of water experts on digitalization. These diciplines are (print screen with the approved curriculum is given below):

- SCADA systems
- Software sensors and BIG DATA
- Automatic control of processes
- Cybersecurity.



Universitatea „Dunărea de Jos" din Galaţi
Departamentul de Formare Continuă şi Transfer Tehnologic
Curs de formare continuă: **Digitalizare în industria apei**
Forma de învăţământ: **cu frecvenţă**
Domeniul de licenţă pe care se fundamentează programul de studii: **Ingineria sistemelor**
Programul de studii universitare de licenţă pe care se fundamentează programul de studii: **Automatică şi Informatică Aplicată**

Competenţe profesionale –Evaluarea prin monitorizare, diagnoză, analiza de date experimentale, în concordanţă cu standarde specifice de performanta a activităţilor de proiectare, implementare, testare, validare, exploatare şi mentenanţă a echipamentelor şi reţelelor de calculatoare folosite pentru conducere automată şi aplicaţii de informatică;

Competenţe transversale - Identificarea rolurilor şi responsabilităţilor într-o echipa plurispecializată, luarea deciziilor şi atribuirea de sarcini, cu aplicarea de tehnici de relaţionare şi muncă eficientă în cadrul echipei; Identificarea oportunităţilor de formare continuă şi valorificarea eficientă a resurselor şi tehnicilor de învăţare pentru propria dezvoltare.

Plan de învăţământ

valabil începând cu anul universitar: 2022-2023

Nr. crt.	Denumire disciplină	Activităţi didactice		Nr. de credite	Forma de evaluare
		C	L		
1.	Sisteme SCADA	3	6	1	V
2.	Senzori software şi BIG DATA	6	12	2	V
3.	Conducerea automată a proceselor	3	6	1	V
4.	Securitate cibernetică	3	6	1	V
Total		15	30	**5**	**4V**
		45			

Below is presented the content developed for the disciplines within the program "Digitalization in the water industry".

SCADA

- course notes –

Table of Contents

1. SCADA systems, model ISA112 (International Society of Automation). SCADA systems architecture.

SCADA systems, an abbreviation for "Supervisory, Control, and Data Acquisition," represent systems for control and data acquisition that have evolved from simple and standalone systems to complex, interconnected systems that communicate through industrial communication networks, sometimes even on a global scale. These systems consist of both software and hardware components.

The development of SCADA systems has progressed from customized systems implemented on specific hardware and software configurations to systems implemented on standard hardware and software platforms. This evolution has led to cost optimization in terms of operation, maintenance, and efficient management due to the multitude of real-time information available in modern systems. However, meeting the increasing demand for information collected from controlled processes to enhance operations and management reveals vulnerabilities in these systems.

Industrial control systems, once closed and utilizing proprietary hardware and software, are now vulnerable to intrusions through external communication networks, including the internet, as well as internal intrusions by personnel serving these systems. Potential attacks exploit vulnerabilities in the standard platforms used in SCADA systems, such as Windows operating systems and PCs, which have been widely adopted within these systems.

The significance of SCADA systems lies in their role as vital components of the critical infrastructures of most nations. They control facilities in the oil and gas industry, transportation, water and wastewater, the energy industry, chemical factories, and many more. SCADA provides real-time data about production operations to enterprise management, implements more efficient control paradigms, enhances facility operation and personnel safety, and reduces operating costs. These benefits are made possible by using standard hardware and software in SCADA systems, combined with improved communication protocols and increased connectivity to external networks, including the internet. However, these benefits come at the cost of increased vulnerability to attacks or erroneous actions from a variety of external and internal sources.

A SCADA system consists of the following three elements:

One or more interface devices with field elements, usually Remote Telemetry Units (RTUs) or Programmable Logic Controllers (PLCs), which gather data from the field—status signals from execution elements and instrumentation elements, execute implemented software programs, transmit commands to execution elements, and send all acquired and processed information to higher-level visualization and monitoring systems.

A communication system consisting of one or more types of industrial communication networks used to transfer data between field data interface devices and the control units and the central element of SCADA. The system can be radio, cable, satellite, etc., or any combination thereof.

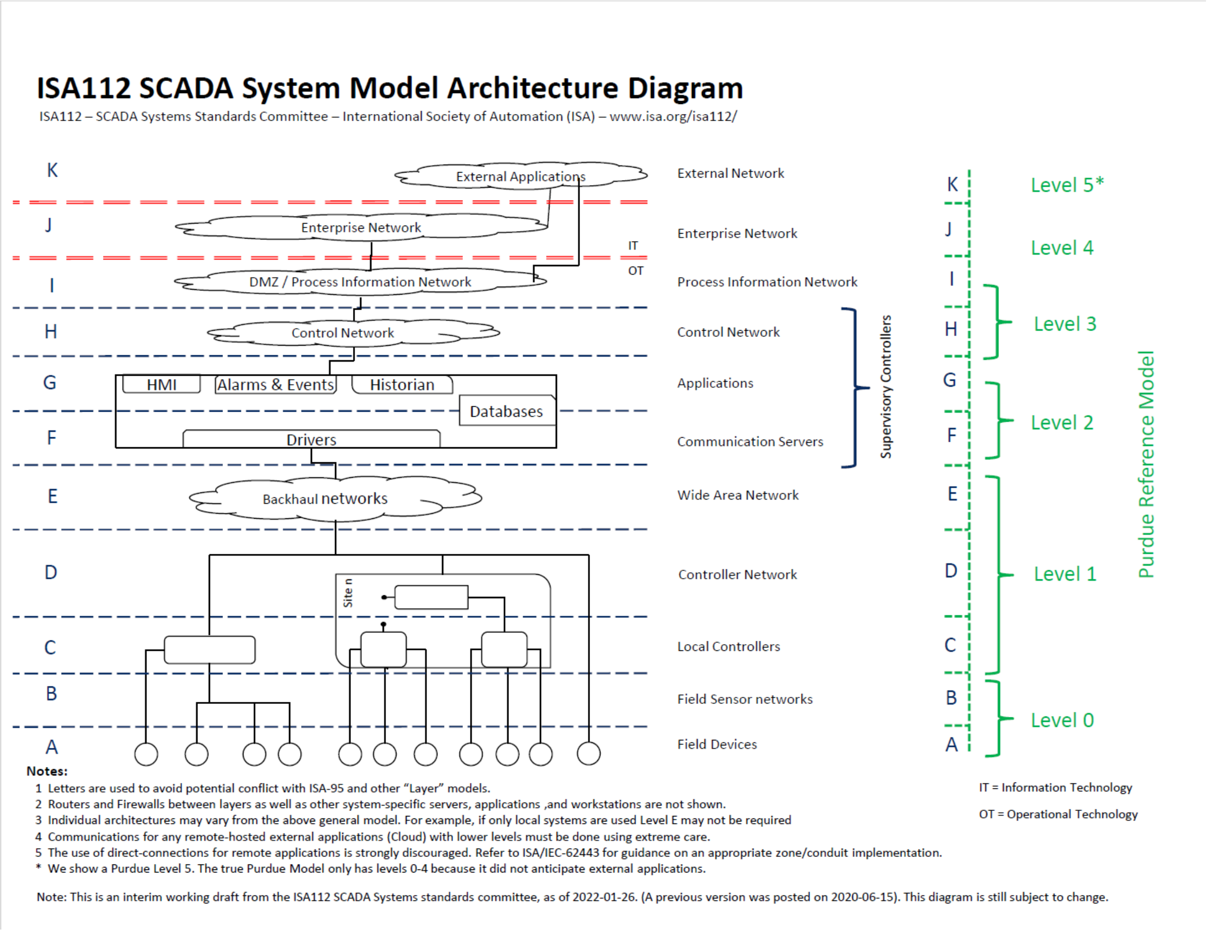
A server or servers or a PLC forming the central element (sometimes called SCADA Center, master station, or main terminal unit).

A collection of standard and/or customized software used in the SCADA center or on field operating interfaces or computers, called Human-Machine Interface (HMI) software.

The SCADA system component is detailed by field specialists' associations aiming to standardize such systems. One of the associations developing standards in the SCADA field is ISA112 – International Society of Automation. The association's committee now has over 200 SCADA experts worldwide, representing a wide range of industries.

The association's activity addresses standardizing the design, implementation, operation, and maintenance of SCADA systems for various industries, essential to support the overall integrity and reliability of these systems. The developing standards and technical reports provide guidance on designing, implementing, operating, and maintaining SCADA systems by documenting best practices across various industries. The anticipated plan is to develop one or more standards complemented by technical reports that expand on implementation details and industry-specific guidelines.

Figure 1 presents the architecture model of a SCADA system in the ISA112 vision.



ISA112 SCADA System Model Architecture Diagram

ISA112 – SCADA Systems Standards Committee – International Society of Automation (ISA) – www.isa.org/isa112/

Notes:
1 Letters are used to avoid potential conflict with ISA-95 and other "Layer" models.
2 Routers and Firewalls between layers as well as other system-specific servers, applications ,and workstations are not shown.
3 Individual architectures may vary from the above general model. For example, if only local systems are used Level E may not be required
4 Communications for any remote-hosted external applications (Cloud) with lower levels must be done using extreme care.
5 The use of direct-connections for remote applications is strongly discouraged. Refer to ISA/IEC-62443 for guidance on an appropriate zone/conduit implementation.
* We show a Purdue Level 5. The true Purdue Model only has levels 0-4 because it did not anticipate external applications.

IT = Information Technology
OT = Operational Technology

Note: This is an interim working draft from the ISA112 SCADA Systems standards committee, as of 2022-01-26. (A previous version was posted on 2020-06-15). This diagram is still subject to change.

Figure 1 –SCADA system model architecture.

The model is a detailed one that expresses the system levels and correlates with another recognized model, the "Purdue Reference Model." The two models are similar, with the ISA model providing a more in-depth breakdown of SCADA system components.

The first ISA112 level is Level A, the level of execution and instrumentation elements (sensors) – "Field Devices." Here, you find actuators, valves, motors, pumps, robots, robotic arms, etc. These are all elements that receive commands from control elements and interact with the controlled process.

The second level, denoted as B, is the sensor network level. Some sensors transmit unified signals to control systems, but there are also intelligent sensors that transmit information over wired or wireless communication networks, such as IO link, DeviceNet, Profibus.

Levels A and B together form Level 0 - "Physical Process" in the Purdue model.

The third level, denoted as C, is the local control equipment level – "Local controllers." In this zone, you find PLCs, RTUs that execute programs controlling the process locally.

The fourth level, denoted as D, is the communication network level between PLCs – "Controller Networks." In this zone, there can be networks like ControlNet, EthernetIP, Profinet.

The fifth level, denoted as E, is the wide-area communication network level that can connect control systems from different locations – "Wide Area Networks." In this zone, there can be networks using protocols like Modbus, etc.

Levels C, D, and E together form Level 1 - "Basic Control" in the Purdue model.

The sixth level, denoted as F, is the server communication level – these are machines running servers with HMI applications, databases, alarms, historians, communication drivers with control level equipment, etc. – "Communication Servers." Some servers may be configured redundantly for process safety.

The seventh level, denoted as G, is the application level – SCADA applications developed containing monitoring and control screens for the installation, alarms, networks, screens for parameters, and all functionality and security configurations, etc. – "Applications."

Levels F and G together form Level 2 - "Supervisory Control" in the Purdue model.

The eighth level, denoted as H, is the control network level – control applications for the controlled system, such as production planning, network management, quality control, MES – "Control Network."

The ninth level, denoted as I, is the DMZ and process control zone level – "Process Information Network."

Levels H and I together form Level 3 - "Operation systems (Manufacturing)" in the Purdue model.

Levels A to I are levels in the Operational Technology (OT) zone.

The tenth level, denoted as J, is the enterprise network zone level – "Enterprise Network." ERP application.

Level J corresponds to Level 3.5 - "DMZ" in the Purdue model.

The tenth level, denoted as K, is the external application level – "External application."

Level K corresponds to Level 4 - "Enterprise (IT)" in the Purdue model.

Levels J and K are levels in the Information Technology (IT) zone.

2. Industrial Communication Networks Used in SCADA Applications..

For two or more systems to communicate, they must speak the same language, known as a communication protocol. Each protocol consists of specific communication rules, including how communication is initiated, how it occurs, and how it is terminated.

SCADA protocols have evolved from the need to send and receive data and control information both locally over short distances and over long distances while maintaining deterministic timing. Deterministic, in this context, refers to the ability to predict the time needed for a transaction to occur when all communication parameters are known. To achieve deterministic communication for critical industrial applications such as refining, electrical utilities, and other SCADA system users, control device manufacturers, such as PLCs, have developed their own communication protocols and structures. Major automation and control equipment manufacturers have developed protocols, including:

Allen Bradley (Rockwell Automation) – a large American company that has developed protocols such as DeviceNet, ControlNet, Data Highway+, Data Highway 485, Ethernet IP, and others.

Siemens – a large European company that has developed Profibus, Profinet, S7 protocol, etc.

Modicon – acquired by Schneider – a European company (originally founded in America) that developed protocols like Modbus RTU and Modbus TCP.

Many of these protocols were initially proprietary, but due to the specialists' concern for standardization beginning in the 1990s, there has been a shift towards developing open protocols for control system communication—non-proprietary standard protocols.

As the Internet gained popularity, companies sought to leverage protocols and tools developed for the Internet, such as the protocol family based on the ISO OSI - TCP/IP model, as shown in Figure 2. Additionally, manufacturers and open standard organizations modified popular and efficient protocols to make them accessible to any communication-capable equipment manufacturer.

A protocol defines the message format and rules for message exchange. High-level models are used to define where protocols are applied and to compartmentalize the functions necessary for sending and receiving messages. The layered architecture model has been widely adopted and is highly efficient. In this model, communication elements are divided into levels with defined interfaces between each level. Two widely used communication reference models are the Open Systems Interconnection (OSI) and the Transmission Control Protocol/Internet Protocol (TCP/IP) model..

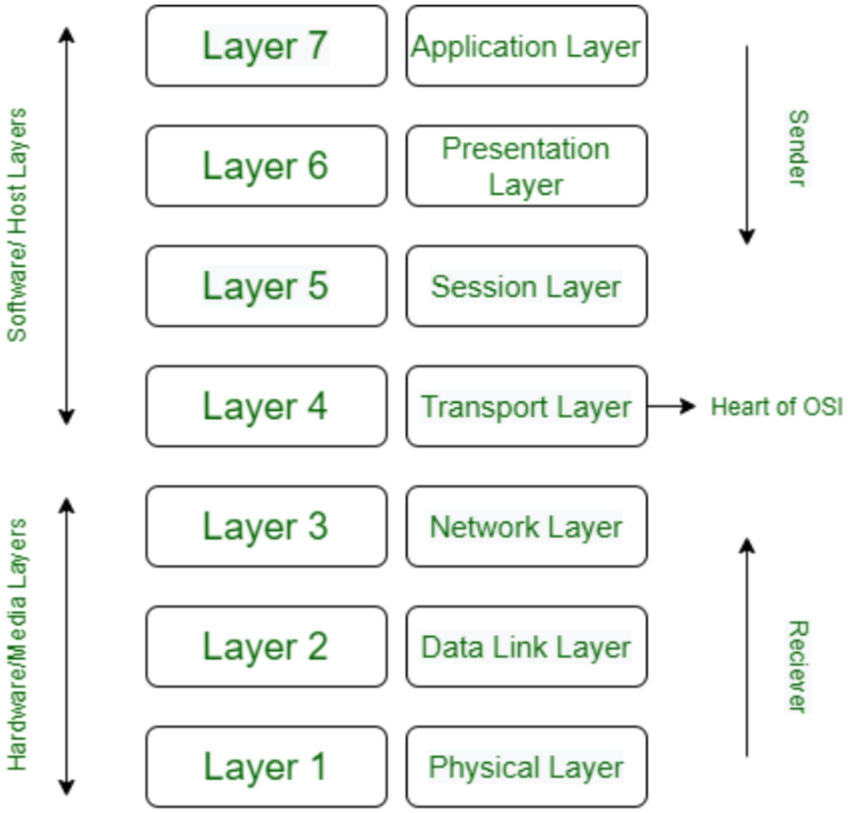


Figure 2 – Model ISO - OSI .

Protocols in SCADA

SCADA system protocols have evolved from proprietary hardware and software specifically designed for SCADA systems. These protocols were developed out of necessity to serve the growing market of real-time computer applications in control situations. In an effort to leverage new network developments, SCADA protocols incorporated versions compatible with internet and local area network technologies. This move led to some standardization but also exposed SCADA systems to attacks commonly used against these technologies in IT environments.

MODBUS Protocols

In the late 1970s, Modicon, Incorporated, developed the MODBUS protocol. MODBUS is positioned at Level 7 of the OSI model and supports client-server communications between Modicon PLCs and other devices on the network. The MODBUS protocol defines methods for accessing a PLC, communication between devices implementing this protocol, and provides means for error detection and reporting. To take advantage of the communication capabilities of instrumentation equipment supporting Ethernet, the Modbus TCP variant was developed. This variant is based on the OSI model, although not all levels of the model are utilized.

DNP3 Protocol

DNP3 is an open SCADA protocol used for serial or IP communications between control devices. It is widely used in applications within water companies and electric power providers for exchanging data and control instructions between main control stations and remote computers or controllers known as remote stations.

Typical commands issued by the main control station include "open valve," "start a motor," and "provide data about a specific control station." The main control station can also provide analog output signals to a remote station. A remote station provides the main control station with information such as pressures, the status of a switch, analog signals representing temperature or power, and information files. DNP3 has also adapted to internet technologies using TCP/IP for DNP3 message exchange.

CAN Protocol

The Controller Area Network (CAN) protocol was developed by Robert Bosch, GMBH, specifically for the automotive industry. It supports serial communications up to 1 Mbps, physically supports up to 110 nodes on a two-wire, semi-duplex network. CAN protocols operate at both Level 1 (physical) and Level 2 (data link layer) of the OSI model.

CAN communications are based on Carrier Sense Multiple Access/Collision Detection (CSMA/CD) as a collision detection method in the network. Multiple devices transmit data over a common bus in this network. When a device senses that the bus is free (no carrier signal on the bus), it attempts to transmit over the bus. In case another device tries to communicate at the same time, devices detect this collision, back off, and try again randomly later. With this approach, specific transmission times across the network cannot be guaranteed. To compensate, CAN offers transmission priorities to nodes using a CSMA/CD + AMP (Arbitration on Message Priority) scheme. CSMA/CD + AMP uses a unique identifier that includes a priority assessment in a message rather than the source and destination node addresses as used in conventional CSMA/CD arbitration. The lower the identifier value, the higher the priority assigned to the message. The length of this identifier varies, being 11 bits in the CAN part A specification and 29 bits for the CAN part B specification.

CIP Protocol

The Common Industrial Protocol (CIP) is an open family of protocols implemented in the application, presentation, and session layers of the OSI model. CIP forms a common upper layer of protocols that can be used above various lower layers, such as those using EtherNet/IP, DeviceNet, and ControlNet, all mentioned below. It also includes a messaging protocol that supports explicit messaging and I/O. CIP is maintained by ControlNet International (CI) and the Open DeviceNet Association (ODVA) provider.

CIP includes communication objects used to define data values, connection type, connection characteristics, and connection timing. It also provides a library of 46 object classes, including control supervisor objects, port objects, identity objects, analog output point objects, parameter objects, discrete input objects, position sensor objects, and AC/DC drive objects.

DeviceNet

DeviceNet is an open standard used to connect equipment such as soft starters, sensors, valve controls, displays, operator interfaces, and higher-level control computers and PLCs. DeviceNet is based on CAN protocols and utilizes the CIP protocol family, including its object libraries and object profiles, for equipment configuration, control, and obtaining data from local devices through CAN protocols to the data link and physical layers. To exchange information, DeviceNet establishes a connection instance using an identity object, a message router object, a DeviceNet object, and a connection object. The identity object contains information such as the device profile, revision number, and provider information. The message router object directs messages to the correct destination, and the DeviceNet object stores lower-level DeviceNet information, such as the MAC ID. The connection object manages the messaging connection. DeviceNet supports communication rates of 125 kbps, 250 kbps, and 500 kbps and can handle a maximum of 64 nodes.

ControlNet

ControlNet is an open real-time network designed for use in SCADA applications. It is a deterministic network that utilizes the CIP protocol object capabilities and can support up to 99 nodes in the network at a data rate of 5 Mbps. It is designed for applications involving multiple controllers and operator interfaces, supporting real-time I/O data exchange as well as messaging information. ControlNet's determinism comes from incorporating the Time Domain Multiple Access (TDMA) concurrent algorithm, allowing a node in the network to transmit data at a specified interval called the network update time (NUT). Critical information is transmitted during an NUT interval, while non-critical information is sent during unscheduled periods when available.

EtherNet/IP

EtherNet/IP protocol also applies CIP by encapsulating CIP messages in Ethernet frames. In addition to the basic CIP object classes, EtherNet/IP uses a TCP/IP object for TCP/IP protocol implementation and an Ethernet link object comprising parameters for establishing an EtherNet/IP link. The CIP protocol operates at the OSI application layer, providing the application object library, at the presentation layer providing message services, and at the session layer supporting routing and connection management. Since Ethernet uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD), which operates by collision detection, attempting to resend communications at random intervals when collisions occur, these communications cannot be called deterministic. This situation poses challenges for real-time data acquisition and control. However, collision reduction methods are implemented, and their impact on communication performance is diminishing, especially with the increased switching speeds, reaching Gigabit Ethernet (10 Gbps). This significantly reduces communication latency. Another factor mitigating this is the option to use User Datagram Protocol (UDP), which is faster than TCP, where the connection is confirmed, and communication is loaded with data for error-checking transmissions. Finally, IEEE developed the 802.1P specification for prioritizing network traffic by incorporating a 3-bit header field that prioritizes messages and allows grouping packets into different classes of priority traffic. Real-time applications, such as motion applications or safety applications, use Ethernet/IP as a communication network between devices.

Profibus

Profibus (Process Fieldbus) is a network standard using an open serial communication principle for critical control and data acquisition applications. It falls under the European international fieldbus standard EN 50 170, defining the functional, electrical, and mechanical characteristics of a Serial FieldBus. Profibus is similar to FieldBus Foundation but offers transmission rates of 31.25 Kbps, 1 Mbps, and 2.5 Mbps at the physical level. As Profibus is an open standard, it can host devices from different manufacturers. Profibus operates at the application, data link, and physical levels of the OSI model. It provides determinism for real-time control applications and supports multimaster and master-slave communication networks.

There are three versions of Profibus, and these are:

- Profibus Process Automation (Profibus PA): Connects acquisition and control devices on a common serial bus. It is also possible for field devices to be powered by the network through the bus. Profibus PA uses the basic functions and extensions available in Profibus DP.

- Profibus DP: Provides high-speed communication between control systems and distributed control devices. Profibus DP has been enhanced by adding diagnostic capabilities, alarm messages, and parameterization, becoming Profibus DPV1.
- Profibus Fieldbus (FMS): Developed to support a large number of applications and higher-level network interconnections between applications at medium transmission rates. It offers a wide selection of functions and is generally more complex to implement than Profibus PA or Profibus DP.

3. Controlling installations using Remote Terminal Units (RTU) and Programmable Logic Controllers (PLC).

RTU systems within SCADA systems have the role of acquiring data and transferring it to level 2, "Supervisory Control." RTU systems are generally PLCs that can be programmed or just configured; they can execute certain tasks in addition to communication. An important requirement for RTU systems in SCADA systems is to retain acquired data for a long period, on the order of hours or days, if communication with level 2 is not functional. A PLC (Programmable Logical Controller) consists of a controller that uses programmable memory to store instructions and implements functions such as logic, sequencing, synchronization, counting, and arithmetic for the control of machines and processes. PLCs are designed to be operated by engineers with limited knowledge of computers and programming languages. The term "logic" is used because programming primarily involves implementing logical and switching operations, for example, if A or B occurs, switch to C; if A and B occur, switch to D. Input devices, such as sensors (e.g., switches), and output devices from the system, such as motors and valves, are connected to the PLC. The operator then enters a sequence of instructions, i.e., a program, into the PLC's memory. The controller then monitors the inputs and outputs according to this program and performs the control rules for which it was programmed. PLCs are similar to computers, but computers are optimized for calculation and display tasks, while PLCs are optimized for control tasks and the industrial environment. Thus, PLCs are::

- Robust and designed to withstand vibrations, temperature, humidity, and noise.
- Have interfaces for inputs and outputs already built into the controller.
- Are easy to program and have a programming language that is easy to understand, primarily focusing on logic and switching operations.

The first PLC was developed in 1969. They are now widely used and control applications ranging from simple ones with 20 digital inputs/outputs to very complex ones with thousands or tens of thousands of digital or analog inputs/outputs. Typically, a PLC system has the basic functional components of a processing unit, memory, power supply unit, input/output interface section, and communication interface.

4. SCADA Software. Functions of SCADA applications. Monitoring, alarming, data acquisition and storage, reporting..

Basic applications used in the development of SCADA applications are primarily created by control equipment manufacturers. Thus, large companies that have developed communication protocols over time have also developed SCADA software alongside PLCs. Some notable examples include:

- Rockwell Automation developed the FactoryTalkView suite of programs and promotes it for creating visualization and SCADA applications.
- Siemens developed WinCC as SCADA application development software.
- Schneider Electric acquired Citect Scada for SCADA application development.

These applications are developed to communicate primarily with their own control equipment, but they increasingly integrate equipment from other manufacturers. Some systems are more closed, making it harder to expand them. However, there is a development trend among all companies to create more open systems that allow the integration of all equipment communicating on standard communication protocols. There are increasingly more protocol conversion devices on the market, making it relatively easy to integrate any standard equipment into any SCADA system today. All these software applications require licensing to be used, and depending on the manufacturer, licensing may be based on the number of screens, tags, servers, clients, tags archived in Historian, redundancy, etc.

Another category of SCADA systems is represented by applications that are not tied to a specific equipment manufacturer but are built to integrate a large number of control systems from various manufacturers. These systems are increasingly developing in the direction of the web, and we have the so-called web-based SCADA systems. One such SCADA application development software based on the web is developed by Inductive Automation and is called Ignition.

This application is multi-platform and runs on Windows, Linux, and Mac OS X. There are implementation options on physical hardware, virtual environments, and managed services; there is even an official Docker Hub image. Ignition can be implemented on:

- Devices: Embedded PCs, laptops, desktops, servers, fog computers
- Managed services: AWS EC2, AWS ECS, AWS Outposts, Azure Virtual Machines, Azure Containers, Google Compute Engine
- Virtual machines and containers: VMware, Parallels, VirtualBox, Hyper-V, Docker

Licensing in Ignition refers to the modules intended to be used in the project. There is a basic configuration that can then be expanded with modules to provide all the necessary functionality. For a licensed application, the number of tags, screens, and clients is unlimited. Also, the number of tags in Historian is unlimited. The important features of this SCADA system, as well as all other systems, include:

1. Data Acquisition.

Ignition SCADA software includes a comprehensive set of data acquisition tools with the OPC UA module built into the base module to connect to almost any type of PLC and the ability to connect to any SQL-type database. Ignition can be configured to save data in any SQL database, resulting in efficient historical data for an industrial process, and it can connect to IIoT devices through the MQTT protocol.

2. Rapid Development of Any SCADA Project

Quick application development under Ignition is facilitated by a powerful Integrated Development Environment (IDE) that provides all the necessary tools. The development environment is integrated into the platform, making it instantly available, running on any major operating system, and comes with an unlimited number of concurrent clients.

3. Real-time Monitoring

Ignition is designed to simplify data transfer, making tag values visible in real-time. The real-time monitoring system in Ignition allows for quickly viewing the status of the installation on any device.

4. Process Control with HMI Devices

Using Ignition, processes can be started and stopped, multiple installations across various locations can be monitored, and the status of the entire factory can be checked at any given time. Ignition includes a design module that allows for easily creating HMI screens optimized to meet all monitoring and control requirements.

5. Visualization Tools.

Dynamic dashboards with powerful data analysis tools can be developed. The system uses a complete library of customizable maps and tables to monitor key performance indicators, view trends at any given moment, and more.

6. Easy Web Deployment with Exceptional Scalability

With Ignition, an unlimited number of clients can be launched instantly without interruptions, throughout the entire runtime, on almost any device connected to a central server. With architectures for almost any type of system and an unlimited licensing model, Ignition can fit any deployment size and can easily grow with the company's needs.

7. Other Features:

- SCADA Alarming: Real-time information about the status of installations in any location.
- Dynamic Reporting: A complete range of dynamic, data-rich reports that can be sent to any location.
- Transaction Management: Easy storage of saved data, stored memory procedures, bidirectional data synchronization.
- Industrial Data History: Developed on an SQL-type database, the historical data saved over time represents a highly efficient module.
- Mobile Access to Applications: Facilitated by the ability to control systems using smartphones and tablets.
- Graphic Symbols: The system contains a library of thousands of customizable graphic elements for use in developed projects.
- SSL Security: Ignition uses SSL to ensure data protection.
- Concurrent Application Design: The ability to develop applications in a concurrent and unlimited environment for development clients.
- Scalability: Easy scaling from one client to the entire enterprise level.
- Critical Systems: Error tolerance is added for critical systems by adding redundant servers.

5. Servers and Clients in SCADA Architectures.

In general, SCADA applications have client-server architectures with redundancy capabilities at the server level. Multiple server-type devices can manage basic database, alarm, HMI, and other applications. The Ignition SCADA system, developed by Inductive Automation, is a web-based SCADA system that includes a set of powerful tools for control and data acquisition (SCADA) - all in one platform that is universal, open, and scalable. Ignition represents a new generation of SCADA systems that address important and challenging issues in SCADA systems. Ignition allows easy control of processes, monitoring, display, and analysis of all data without limitations.

SCADA system architectures in Ignition are diverse, ranging from simple to complex.

1. Simple architectures. These can be with or without redundancy. The following figure illustrates this type of architecture.

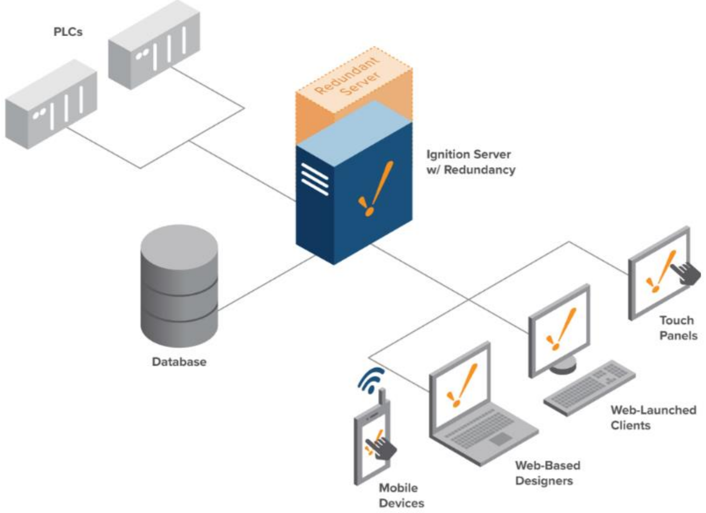


Figure 3 – Simple SCADA Architecture with Redundancy.

The basic architecture with redundancy is most suitable for applications that require a scalable SCADA system, centrally managed, using a single on-premise Ignition server (with a redundant server) and specific modules. This implementation is the most cost-effective configuration. Unlimited connections, tags, databases, and web clients are available. However, the architecture can also be without redundancy, with the Ignition server - Gateway - running on a single machine..

2. Scale-Out Architectures with Redundancy.

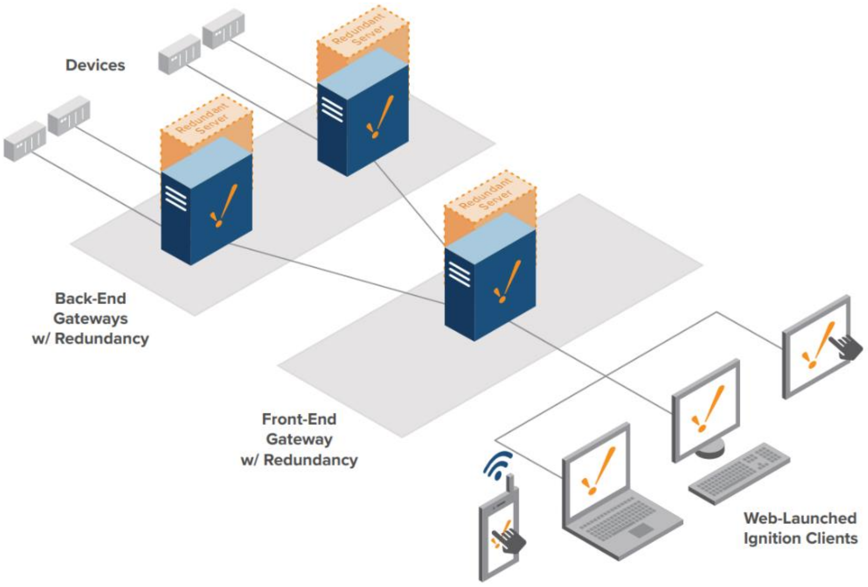

Devices
Redundant Server
Redundant Server
Redundant Server
Back-End Gateways w/ Redundancy
Front-End Gateway w/ Redundancy
Web-Launched Ignition Clients

Figure 4 – Scale-Out SCADA Architecture with Redundancy.

With the Scale-Out architecture, tasks are divided between back-end gateways (with redundant servers) handling device data and front-end gateways managing client applications. This architecture easily scales without overloading a single gateway..

3. „ Hub & Spoke" Architecture.

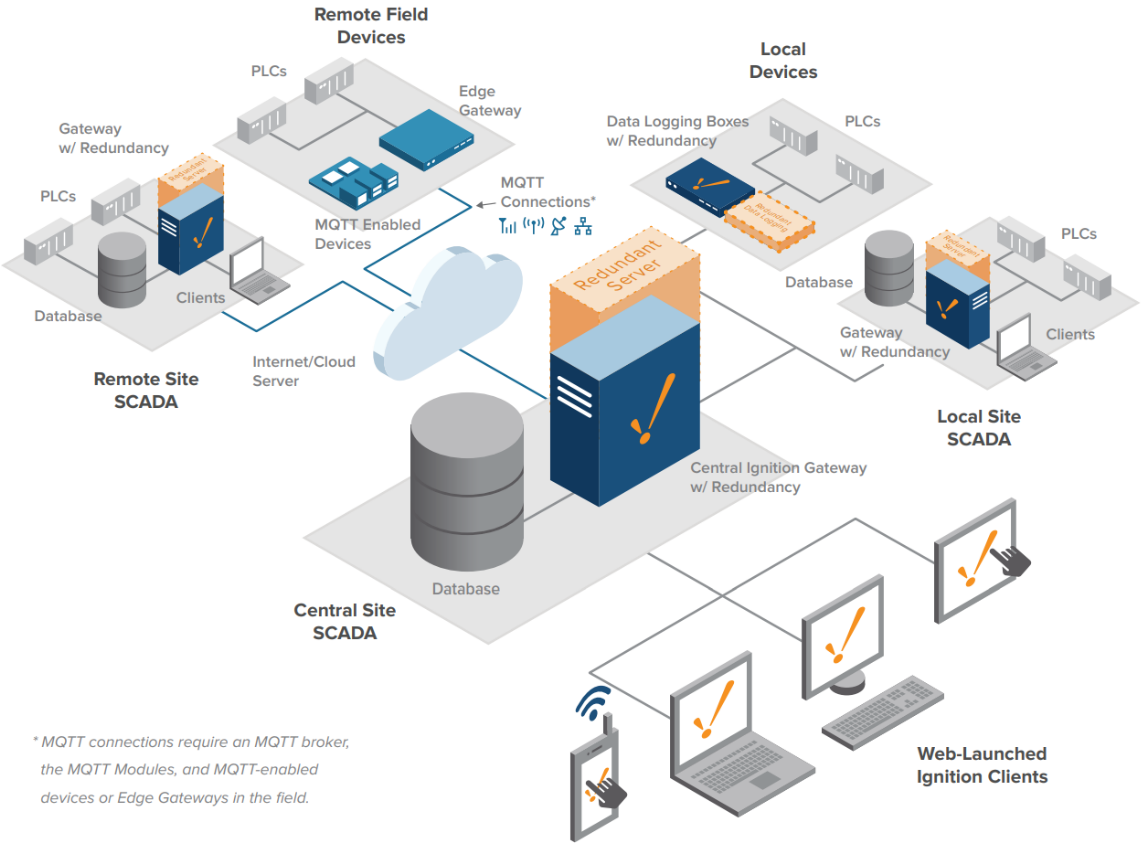


Figure 5 – "Hub & Spoke" SCADA Architecture with Redundancy.

The "hub and spoke" architecture consists of two components. The "HUB" component consists of a central Ignition Gateway (with redundancy) with Vision, Reporting, and Mobile modules, and a database server. The "SPOKE" component consists of an Ignition Gateway (with redundancy) with OPC UA and SQL Bridge modules dedicated to data recording. Each site is entirely independent, operating with its own history, alarms, and clients, with the client gateway used for coordination and long-term history storage. In the event of a communication failure between the central and remote servers, the data is stored at the remote site until communication is restored. Once communication is restored, the data that needs to be saved in the long-term history is automatically synchronized. It is noted that a remote installation communicates with the central gateway via the MQTT protocol, which is a communication module.

4. "Enterprise" Architecture.

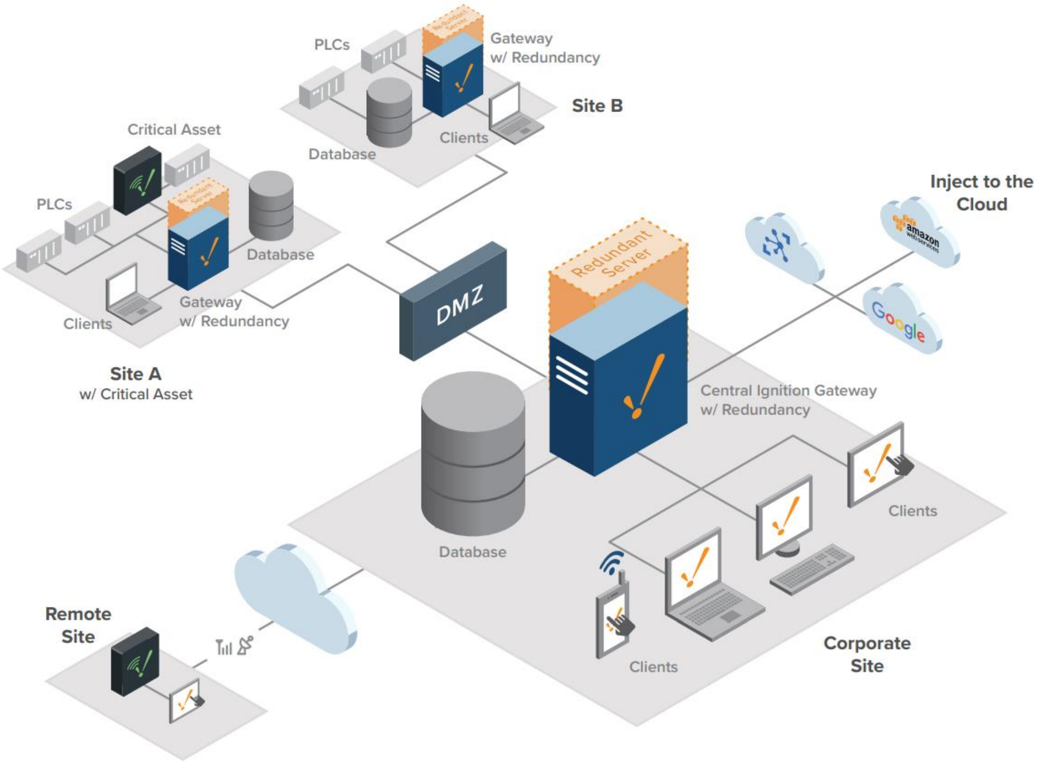


Figure 6 – "Enterprise" SCADA Architecture with Redundancy.

The Enterprise Ignition architecture (with redundancy) allows us to create a connected and secure system on multiple levels. Multiple sites containing critical data connect to a central company server. Using Ignition Edge ensures that data from critical assets is never altered. Connecting through a DMZ provides an additional level of security for data transfer and access. The Ignition Gateway easily connects to cloud services such as Microsoft Azure, AWS, IBM Cloud, and Google Cloud for storage and analysis.

5. "IIOT" Architecture

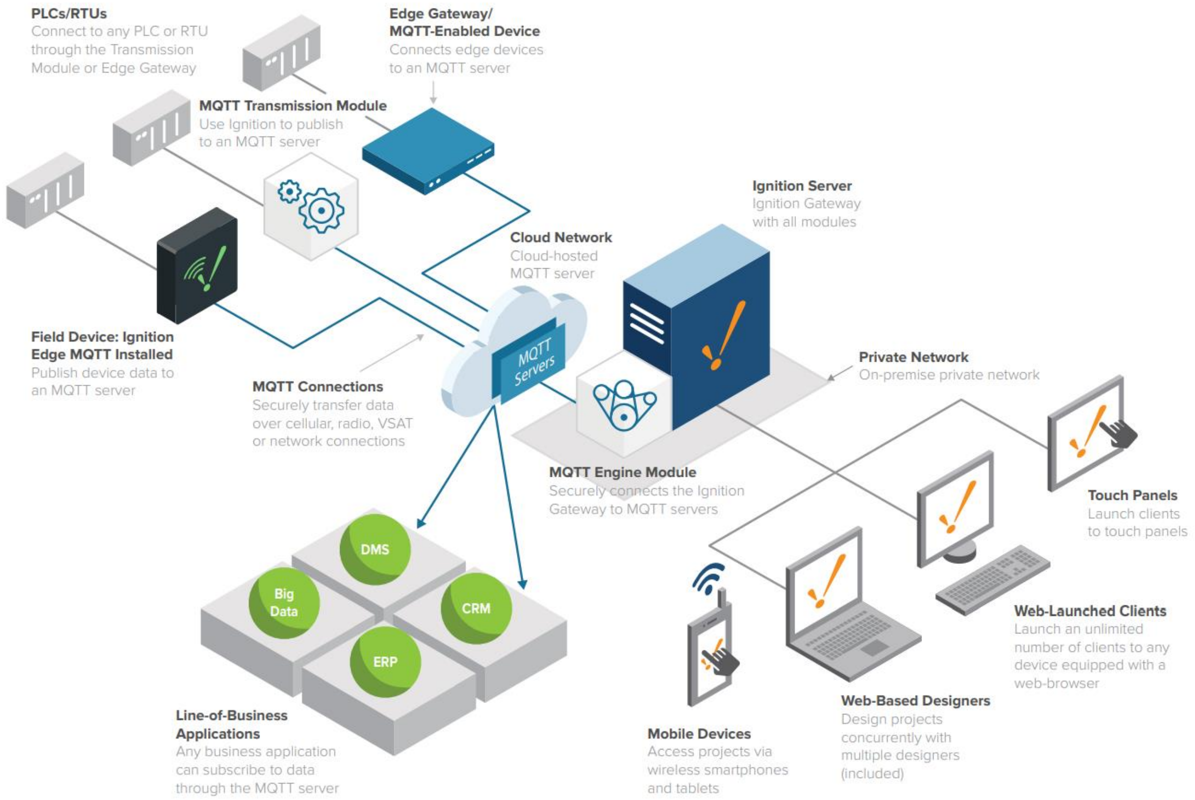

PLCs/RTUs
Connect to any PLC or RTU through the Transmission Module or Edge Gateway
Edge Gateway/ MQTT-Enabled Device
Connects edge devices to an MQTT server
MQTT Transmission Module
Use Ignition to publish to an MQTT server
Ignition Server
Ignition Gateway with all modules
Cloud Network
Cloud-hosted MQTT server
MQTT Servers
Field Device: Ignition Edge MQTT Installed
Publish device data to an MQTT server
MQTT Connections
Securely transfer data over cellular, radio, VSAT or network connections
Private Network
On-premise private network
MQTT Engine Module
Securely connects the Ignition Gateway to MQTT servers
Touch Panels
Launch clients to touch panels
DMS
Big Data
CRM
ERP
Web-Launched Clients
Launch an unlimited number of clients to any device equipped with a web-browser
Web-Based Designers
Design projects concurrently with multiple designers (included)
Line-of-Business Applications
Any business application can subscribe to data through the MQTT server
Mobile Devices
Access projects via wireless smartphones and tablets

Figura 7 – "IIOT" SCADA Architecture.

Ignition IIoT can collect data from any device at the network edge, publish that data to a central broker, and transmit it to subscribing industrial and business line applications.

Ignition IIoT can connect to field PLCs using the MQTT transmission module, field devices with installed Ignition Edge MQTT, and/or MQTT-enabled Edge gateways and field devices using the Cirrus Link Sparkplug MQTT specification.

This data is published to an MQTT broker, which can be located on-premise, in the cloud, or in a hybrid configuration between the two.

The MQTT Engine module located on an Ignition Gateway can subscribe to any data published by the broker, and this data can be used in any Ignition application..

6. „ Cloud Hybrid” Architecture.

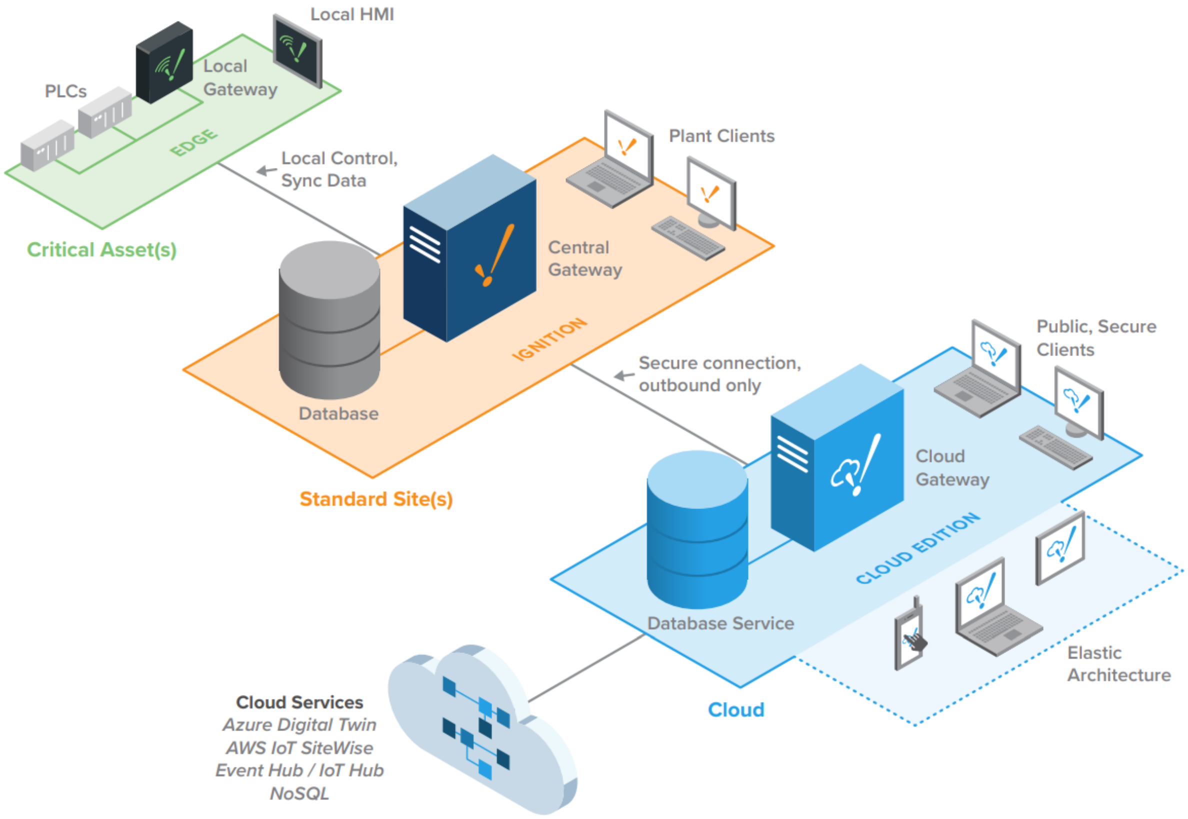


Figure 8 – "Cloud-Hybrid" SCADA Architecture.

The "Cloud-Hybrid" architecture allows us to secure and distribute data at any level. The use of Ignition Edge ensures that data from critical assets is never compromised, while the use of standard Ignition allows connection to multiple sites at a central corporate server.

Ignition Cloud Edition is used to build flexible enterprise architectures that expand on-demand using public or private cloud services.

Since AWS and Azure host Ignition Cloud Edition, users are not responsible for maintaining the cloud server hardware; this means additional storage and cloud computing power can be provided and freed up on-demand, allowing for the development and rapid deployment of enterprise applications.

6. SCADA Systems Security – Security Threats, Solutions for Ensuring Security in SCADA Systems. NIS Directive, Law 362/2018.

Most networks, including SCADA system networks, face common security issues and require appropriate control elements. An important consideration for SCADA networks is that they cannot afford non-deterministic delays, security mechanisms requiring large memory capacities, operator lockouts, and relative long processing times. However, some of the fundamental security measures available for SCADA systems are similar to those used for OSI and TCP/IP Layered Architectures. Network best practices include protecting the confidentiality, integrity, availability of data, along with providing authentication and access services.

In Romania, Law no. 362/2018 regarding ensuring a common high level of security for networks and information systems came into force, transposing the so-called NIS Directive (Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of networks and information systems across the Union). Its purpose is to increase the preparedness of EU states to deal with cybersecurity incidents and, consequently, enhance citizens' trust in the Digital Single Market.

This specifically addresses:

1. Operators of Essential Services (OSE) in 7 sectors of economic activity:
   I. Energy
   II. Transport
   III. Banking sector
   IV. Financial market infrastructures
   V. Healthcare sector
   VI. Supply and distribution of drinking water
   VII. Digital infrastructure
2. Digital Service Providers (DSP) in three categories, namely: online marketplaces, online search engines, and cloud computing services.

Security Law Establishing Measures and Requirements for Effective Security and the Obligation to Notify Incidents.

To ensure a common level of security for networks and information systems, operators of essential services and digital service providers are obligated to adhere to the technical standards developed by CERT-RO. CERT-RO, in consultation with the authorities regulating the specified sectors and sub-sectors, develops guidelines to support the implementation of minimum security measures for operators and providers of essential services.

The technical standards applicable to operators of essential services are established based on at least the following categories of activities for ensuring the security of networks and information systems:

1. Access rights management.
2. User awareness and training.
3. Logging and ensuring traceability of activities within networks and information systems.
4. Testing and evaluating the security of networks and information systems.
5. Configuration management of networks and information systems.

6. Ensuring the availability of essential services and the functioning of networks and information systems.
7. Continuity management of essential service operations.
8. Identity and user authentication management.
9. Incident response.
10. Maintenance of networks and information systems.
11. Management of external memory supports.
12. Ensuring the physical protection of networks and information systems.
13. Development of security plans.
14. Ensuring the security of personnel.
15. Risk analysis and assessment.
16. Ensuring the protection of products and services related to networks and information systems.
17. Vulnerability and security alert management.

Some of the security measures applicable to SCADA systems are described below.

Firewall-uri

A key security protection element necessary for any network connected to an untrusted network, such as the Internet, is a firewall. A firewall provides protection against viruses, worms, and other types of malicious code, as well as network intrusions. One issue with firewalls applied to SCADA systems is that most firewalls do not support the management of SCADA protocols. This situation is being investigated by various organizations, and some SCADA-aware firewalls are under development.

Proxy Firewalls

Proxy or application-level firewalls operate at the 7th layer of the OSI model. In the dictionary, a proxy is defined as a person authorized to act on behalf of another; an agent or substitute. Thus, proxy software can be placed between a user and a server to hide the user's identity. The server sees the proxy and cannot identify the user. The scenario is valid in reverse, where the user interacts with the proxy software in front of the server, and the server or its associated network cannot be identified. A proxy firewall is effective in shielding a network from an untrusted external network, such as the Internet.

Demilitarized Zone (DMZ)

Firewalls can be used to implement security network architectures that are effective for SCADA systems. These architectures are based on the concept of a demilitarized zone or DMZ. A DMZ is a region that provides separation between an external or public network and an internal or private network. For a firewall to support a DMZ, it must have multiple external interfaces and appropriate access control lists where needed. Several different architectures use DMZs, but two are particularly applicable in acquisition and data control environments. These architectures are a single firewall DMZ and a dual firewall DMZ. They can serve the purpose of separating a corporate enterprise network from the control network while providing a connection for both to a public network, such as the Internet.

Single Firewall DMZ

In a single firewall DMZ, a firewall is used to filter data packets from, for example, a corporate network to the local control network and from an external network to the corporate network. The DMZ contains elements that need to be accessed by enterprise computers, as well as the connection to the external public network. This architecture is shown in Figure 3-16. Because there is no firewall between the DMZ and the control network, the control network is potentially vulnerable if the DMZ is penetrated by an attack from the external network or through the enterprise network.

7. Bibliography

a. International Society of Automation: https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa112
b. https://www.geeksforgeeks.org/how-communication-happens-using-osi-model/
c. https://inductiveautomation.com/ignition/architectures
d. Directoratul National de Securitate Cibernetica: https://dnsc.ro/pagini/ansrsi
e. Flexible Solutions for Your Supervisory Control and Data Acquisition Needs - Rockwell Automation Publication AG-SG001G-EN-P - April 2015
f. SCADA System - Application Guide - Publication AG-UM008C-EN-P - February 2005
g. Securing SCADA Systems, Ronald L. Krutz, Wiley Publishing, Inc.

Practical activity 1

Selection of Automation System Equipment.

GIVEN:

Technological engineers and hydraulics specialists will provide a P&ID diagram, a list of motors and transducers, and a control philosophy for a newly designed installation.

The installation's scheme is presented in figure 1.

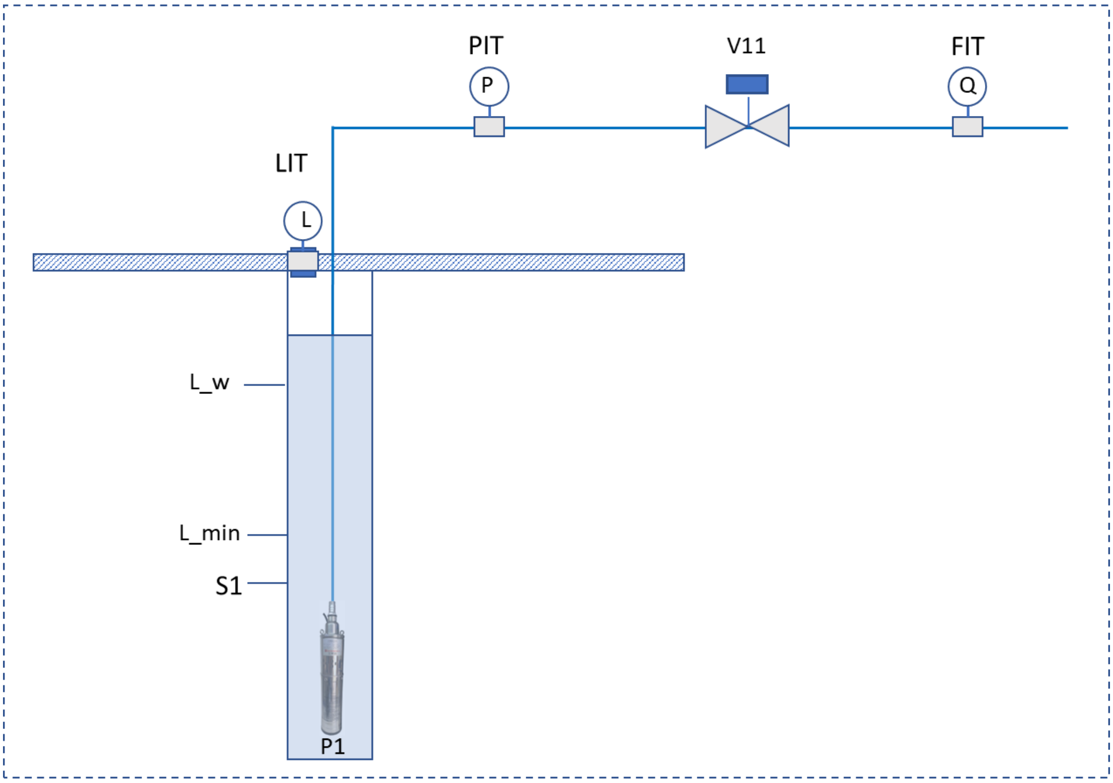


Figure 1 – Raw Water Drilling Installation

The operation of the installation is described below:

The installation represents a raw water drilling system, extracting water from the groundwater, which is then transferred to a water treatment station.

Raw water is extracted from a well drilled to a depth of 30 meters. The extraction of raw water is done with a 5 kW submersible pump installed in the well, which is operated by a frequency converter.

The well is equipped with a level transducer (LIT1) that measures the water level in the well. When the level drops below a parameterized level set from the operator interface, with an initial value of 2 meters, the pump stops. After the pump stops due to the decrease in the well's water level, it waits until the level is restored to a operating threshold at which the pump will start. This threshold is also parameterized and has an initial value of 18 meters. For safety, the pump's software thresholds are duplicated by a float-type safety sensor that will provide a digital signal - S1. It will be mounted slightly below the level set as a limit for the value read from the transducer.

On the pipe that transports water from the well to the treatment station, a pressure transducer in the range of 0 to 10 bars is provided, which provides a 4-20 mA signal. After the pressure transducer, an isolation valve, V11, is provided. A digital output is required to control it, as well as two inputs for open and closed confirmations of the valve. After the isolation valve, a flow meter is installed, which will provide information on the instantaneous flow rate and will display the flow rate totalizer through the well.

REQUESTED:

Design an automation system with PLC for the given installation using the IAB (Integrated Architecture Builder) application presented in the course. The following additional requirements will be observed:

1. Create a table in Excel or a similar program with the objects and components of the installation, including the number and type of I/O required for the control of the installation.
2. Based on the number of I/O, create a project in IAB for a system with Micro800 PLC.
3. Use additional plugin and expansion slot inputs.
4. Add a 10" PV800 operator interface.
5. Connect to the PLC via an EthernetIP network through an unmanaged 8-port Stratix switch.
6. The pump will be controlled through a PowerFlex 525 frequency converter connected via EthernetIP.
7. Include an ET1000 energy parameter measurement unit connected via EthernetIP.
8. Provide a power supply for the equipment in the automation cabinet.

Objectives

1. Calculation of the number of input and output signals required for the control of the given installation.
2. Creation of a list of automation equipment necessary for the realization of a PLC automation panel that controls the proposed installation and adheres to its functionalities.
3. Utilization of the IAB software application for the configuration and validation of a control system for the given installation.

Practical activity 2

Development of a PLC Application for the Given Installation

GIVEN:

1. List of automation equipment necessary for the realization of a PLC automation panel that controls the proposed installation and adheres to its functionalities.
2. Excel table or a similar program with the objects and components of the installation, including the number and type of I/O required for the control of the installation..

REQUESTED:

Develop an application for the PLC resulting from Practical Activity 1, respecting the following requirements:

1. The PLC program will be developed in CCW (Connected Components Workbench) using the Ladder Logic language and will be named AP2_data.
2. Configure the PLC from the equipment list with the necessary modules.
3. Assign a private IP address to the PLC from class C: 192.168.0.10.
4. Create the tag list of the project based on the table developed in AP1 and according to the design data. Import the tags into the CCW application.
5. Implement local variables that signify operating conditions.
6. Scale analog quantities such as level, pressure, and flow in both electrical (mA) and physical sizes characteristic of each transducer.
7. Implement the general alarm button.
8. Implement the system's operation in two modes - MANUAL and AUTOMATIC, imposed by a software key on the operator interface.
9. In MANUAL mode, the execution elements will operate as follows: pump P1 will start at a fixed frequency set by the operator only if isolation valve V1 is open; isolation valve V1 will open and close through the action of two software buttons on the operator interface - Close and Open.
10. In AUTOMATIC mode, the following control algorithm for the drilling system will be implemented: when pressing an AUTOMATIC START button, visible only in this Automatic mode, pump P1 will start if it meets the start conditions - level higher than a threshold and if valve V11 is open.
11. The start sequence unfolds as follows: - Check the level in the well, and if it is higher than L_min, then open valve V11, and after it confirms that it is open, pump P1 will start. The frequency at which the pump starts is 40 Hz.
12. The stop sequence unfolds as follows: - if the well level has reached the minimum level, then stop the drilling pump, then close valve V11. Wait until the well level reaches the L_W value, and then resume the start sequence.

13. When pressing an AUTOMATIC STOP button, the drilling pump will stop, and then valve V11 will close.

14. Record the operating time for the drilling pump.

Objectives

1. Designing a software application for the Micro800 series PLC.
2. Using CCW for its development.
3. Implementing this application in the ladder logic programming language.

Practical Activity 3

Elaborarea unei aplicații HMI pentru instalația data

GIVEN:

1. The application developed in Practical Activity 2..

REQUESTED:

Develop an application for the PanelView 800 - HMI (Human-Machine Interface) that adheres to the following requirements:

1. In the project containing the PLC program developed in CCW (Connected Components Workbench) in Practical Activity 2, add a PanelView 800 10".
2. Configure the PanelView to communicate with the PLC in the project.
3. Import all tags from the PLC application into the HMI application.
4. Set default properties for the screens to be developed.
5. Create a total of four screens. Each screen will contain navigation buttons that allow navigation between screens. Each screen will have a title, date, and time displayed at the top.
6. The first screen will be an introduction screen displaying information about the installation, the operating status, and buttons for selecting Automatic and Manual modes, as well as the Start Installation button. The operating conditions will be displayed, and the operating mode will be shown on each screen.
7. The second screen will contain the drilling installation with all its components. Here, buttons for opening and closing the valve, Start and Stop for the pump, and setting the frequency in both manual and automatic modes will be displayed. The physical values provided by the pressure, level, and flow transducers will also be shown. The flow counter will be displayed.
8. The third screen will contain parameters for scaling analog signals. Scaling parameters will be entered, and the values provided by the transducers in CAN signals, unified signal, and corresponding physical value will be displayed.
9. The fourth screen is dedicated to alarm signals.

Objectives

1. Designing a software application for the PanelView800 HMI.
2. Using CCW software for its development.

SOFTWARE SENSORS

- course notes –

I. Introduction

Due to climate changes in the last decade, especially the presence of drought in Romania, water availability has become a concerning issue due to the increasing need for irrigation of agricultural areas by farmers and others, as well as the demand from a growing population and industrial development. Considering these requirements, water resource optimization involves treating wastewater to be discharged into the natural cycle.

The mission of treating wastewater is by no means a simple task, as this process must adhere to the standards regulated by the European Union regarding water quality. The wastewater treatment process requires suitable instrumentation capable of generating relevant information [1] that reflects the accuracy of variables that are continuously changing, especially due to climate changes. The existing instrumentation in the literature involves the use of hardware sensors and is of three types: online, offline, and in-line instrumentation. The information provided by sensors is of real importance both for water quality and for controlling the entire water treatment process.

Although the use of sensors is a modern and convenient solution for monitoring the water treatment process, special attention must be given to the lifespan and wear and tear of these sensors. Considering the economic situation and the difficulty in identifying hardware sensor malfunctions, finding solutions to improve sensor reliability or replacing hardware sensors with software sensors is essential.

II. Monitoring Process Parameters in Wastewater Treatment Plants via Software Sensors

Monitoring parameters in purification processes is a fundamental component in the automation of wastewater treatment facilities. The main purpose of monitoring is to achieve optimization of the entire wastewater treatment process. By continuously monitoring key parameters, the following objectives are pursued:

- **Minimizing time and resource losses**: Continuous evaluation of parameters allows for the rapid detection and correction of deviations from ideal parameters, thereby reducing time and resource losses in the purification process.
- **Maintenance and repair planning**: Constant monitoring provides essential data for planning maintenance and preventive repairs. This helps avoid major malfunctions and keeps the facility in the best operational condition.
- **Energy efficiency**: By monitoring and adjusting parameters in real time, the energy consumption of the facility can be optimized. This leads to a significant reduction in operational costs and substantial long-term energy savings.

In conclusion, monitoring parameters in wastewater treatment processes is not only an essential activity but also an intelligent investment for optimizing the purification process, reducing losses, and increasing efficiency in resource utilization.

III. Definitions and Benefits of Software Sensors

- **Sensor** – is a device that measures a physical quantity (pressure, light, temperature, humidity, etc.) and transforms it into a signal that can be read by an observer through an instrument [2].
- **Software sensor** – the term combines the words "software" because signal evaluation models of the sensor are usually computer programs, and "sensor" because these models provide information as hardware sensors would [3]. This means that the signal is produced by software instead of hardware sensors (see Figure 1).

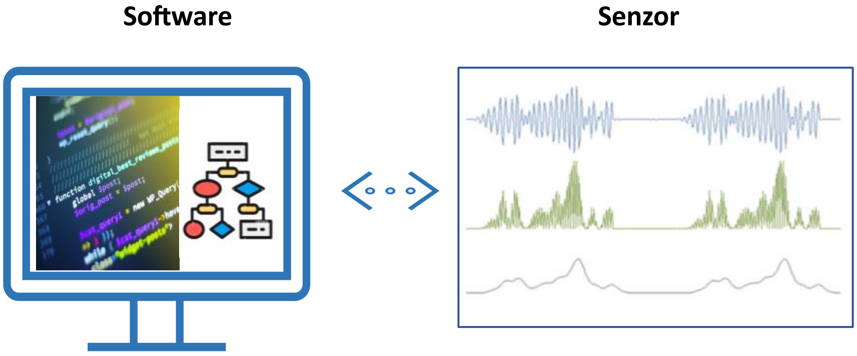


Fig. 1 Software sensor graphical representation

Wastewater treatment plants utilize a variety of sensors to monitor and control various parameters during the wastewater treatment process. These sensors contribute to ensuring efficient operation and the quality of treated water. A classification of sensors used in the water treatment process is illustrated in Figure 2, but among the most commonly encountered are:

- Level Sensors: These sensors monitor the water level in various parts of the treatment plant, such as tanks, sedimentation basins, or collection channels.
- Flow Sensors: Flow sensors measure the quantity of water entering or leaving different stages of the treatment process. This data is essential for controlling water flows.
- Dissolved Oxygen Sensors: Dissolved oxygen in water is crucial for the survival of bacteria that break down organic matter in the treatment process. Dissolved oxygen sensors help monitor the real-time oxygen concentration.
- pH Sensors: The water's pH is an important indicator for controlling chemical processes in the treatment plant. pH sensors ensure that the pH value remains within the appropriate range for necessary chemical reactions.
- Turbidity Sensors: Turbidity measures how clear or cloudy the water is. Turbidity sensors help assess water quality and monitor the effectiveness of the purification process.
- Temperature Sensors: Water temperature can affect microbiological and chemical processes in the treatment plant. Temperature sensors ensure monitoring of this variable.
- Chemical Substance Sensors: Depending on the requirements, treatment plants may use specialized sensors to measure specific concentrations of chemical substances, such as chlorine or toxic substances.
- Solid Level Sensors: These sensors measure the level of solid materials (e.g., sand or sediment) in water, assisting in managing sediments and waste.
- Gas Sensors: Some treatment plants may use sensors to detect gases present during the treatment process, such as hydrogen sulfide (H2S), which can be toxic.
- Methane Gas Sensors: These sensors can be used in anaerobic digestion processes to monitor methane gas production..

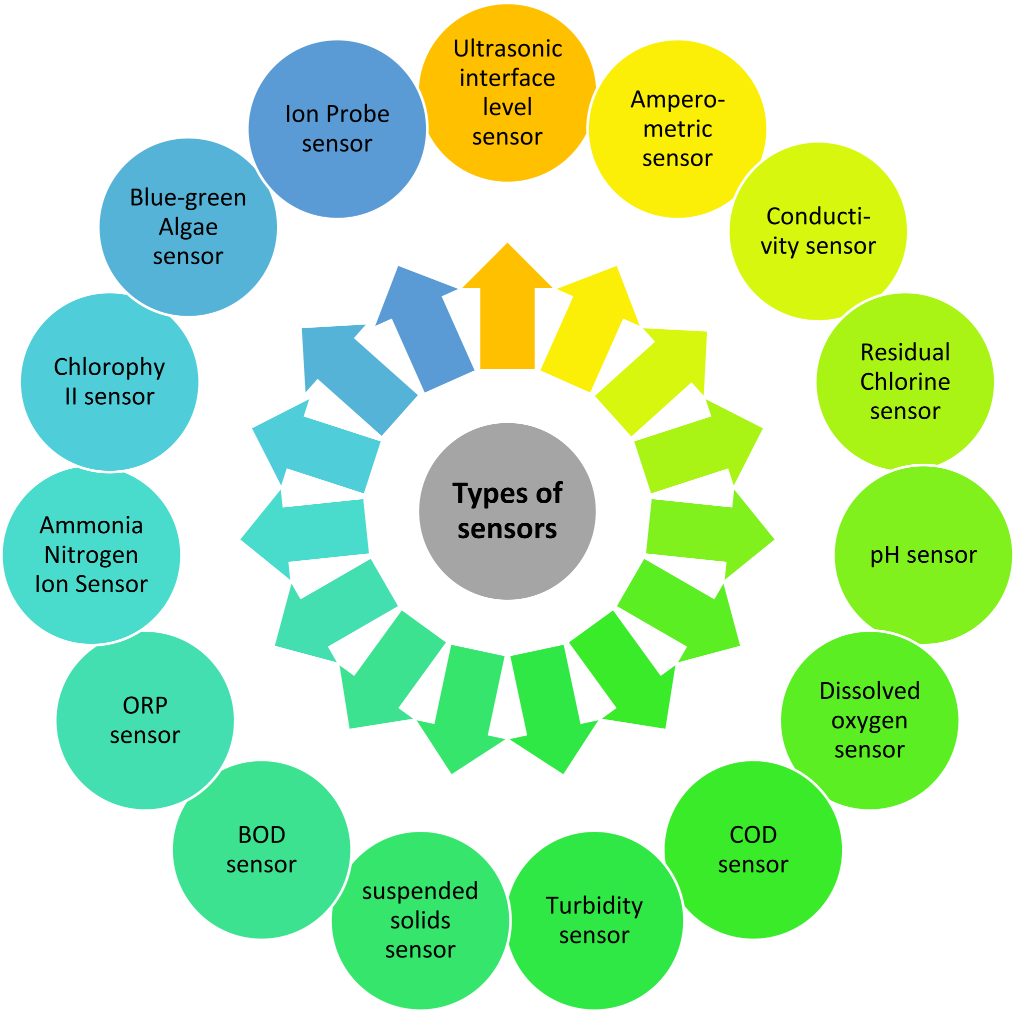


Fig. 2 Types of sensors used in water industry

What are software sensors in the wastewater industry?

Software sensors in the wastewater industry refer to digital tools or applications that use software algorithms to monitor and analyze data related to wastewater treatment processes. These sensors utilize data-driven models and control strategies to provide information and optimize the operations of wastewater treatment plants.

Advantages of using software sensors in the wastewater industry:

- Real-time Monitoring and Control: Software sensors enable real-time monitoring over the internet of critical parameters such as water quality, water level, pressure, and consumption. These advantages ensure the rapid detection of issues and allow for immediate action.

- Cost Reduction: Software sensors eliminate the need for replacement due to wear and tear or maintenance of wastewater treatment facilities.

- Operational Efficiency: Real-time data from software sensors facilitates process optimization, leading to energy, water, and resource savings and contributing to more efficient operations.

- Flexibility: Software sensors allow for customized configuration and can be adapted to the specific needs of wastewater treatment facilities.

- Scalability: The ability to replace and add sensors without the need for modifications to wastewater treatment facilities.

- Rapid Intervention: Software sensors detect deviations from normal parameters and can trigger alarms, allowing for quick intervention to prevent potential breakdowns or leaks.

- Easy Access to Data and Remote Management: Collected data can be easily accessed over the internet without the need for additional hardware units.

- Software Updates: Sensors can be improved by adding new functionalities or addressing issues through software updates.

- Prediction: Software sensors provide the capability to deduce measurements of immeasurable variables.

- Reliable Calculation of Parameters in the Absence of Hardware Sensors.

- Risk Reduction of Bioreactor Contamination.

IV. Software sensor models

The general methodology for developing a software sensor involves several stages, such as:

- Data collection,
- Data inspection,
- Selection of historical data,
- Data preprocessing,
- Selection of a mathematical model,
- Training and validating the model,
- Integration of the software sensor,
- Monitoring and maintenance of the software sensor.

In general, there are three models for software sensors (see Figure 3) [4]:

- Model-driven,
- Model data-driven,
- Model gray-box.

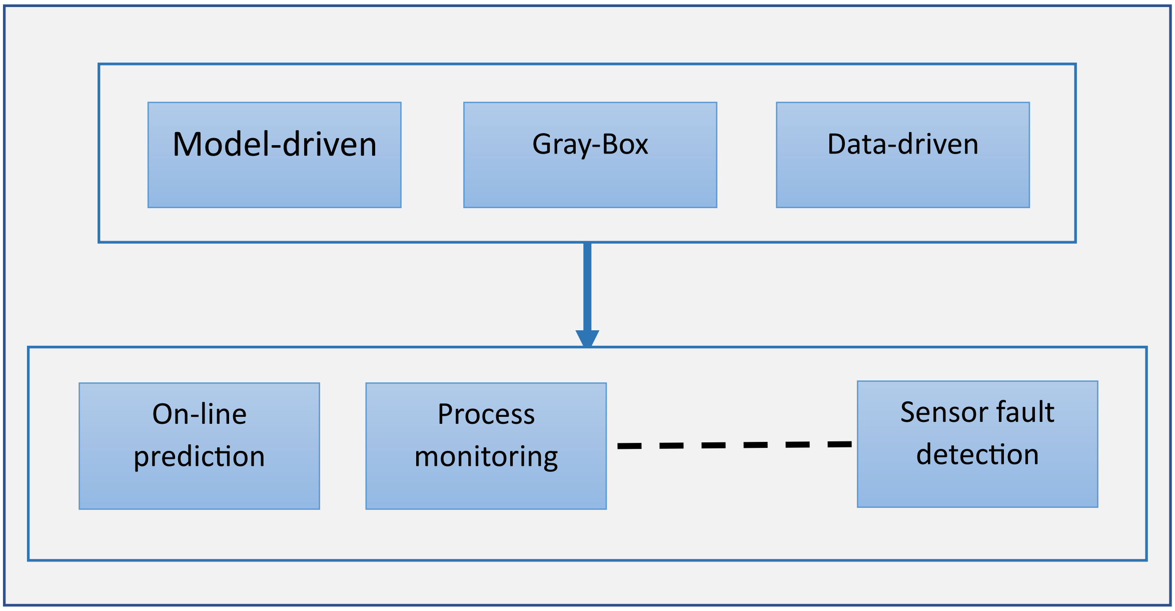


***Fig.* 3** Software sensors models [4]

Model-driven sensors (or white-box) rely on complete phenomenological knowledge of the process [4]. These devices operate based on mathematical models and algorithms. These sensors are designed to measure, monitor, and collect data according to predefined mathematical models or specific algorithms.

Data-driven sensors (or black-box) rely on historical data [4] and are devices that collect data and provide results without revealing details about how the internal process or algorithm works. These sensors are focused on collecting and providing data without exposing the process of generating this data.

Gray-box sensors are a combination of model-based and data-based software sensors [4]. These sensors provide a blend of features, offering both accurate data and controllable transparency regarding their internal operation.

V. Aplicabilitatea senzorilor în procesul de tratare a apelor uzate

Regarding the interest in the applicability of sensors in the wastewater treatment process, it is heightened because collecting a large amount of data provides human operators with the capability to anticipate decision-making. Thus, in the approach [5], an alternative cost-effective method for monitoring essential parameters of wastewater quality, such as TP (Total Phosphor) and COD (Chemical Oxygen Demand), was tested and validated in a large-scale wastewater treatment plant. In [6], the efficiency of a method for detecting sensor faults in DO (dissolved oxygen), based on Principal Components Analysis (PCA), was demonstrated. Satisfactory results were reported in [7], where the efficiency of soft sensor learning techniques for complex phenomena to predict ammonium was demonstrated. For the prediction of effluent chemical oxygen demand (COD) and total nitrogen (TN) with large variations, a machine learning model based on IFFNN coupled with genetic algorithm was developed in [8]. In estimating nutrient concentrations in effluents in a biological wastewater treatment plant, a hybrid learning method combining genetic algorithm with adaptive neuro-fuzzy inference system (GA-ANFIS) was applied [9]. The results indicate that the hybrid GA-ANFIS soft sensors outperform ANFIS-based soft sensors in terms of effluent prediction accuracy [9]. However, in the study [10], multi-output soft sensors were developed using the multivariate linear regression model (MLR), multivariate relevance vector machine (MRVM), and multivariate Gaussian process regression (MGPR).

The proposed method was validated by applying the algorithm to a wastewater treatment plant simulated with the BSM1 model. Additionally, in [11], the results of the neural network-based MLP (multilayer perceptron) model are analyzed, providing a better estimate than the corresponding MLR (traditional multiple linear regression). For soft-sensor modeling of effluent COD, TN, and TP concentration in a municipal activated sludge process, Kim et al. [12] used MPCA and FFNN.

VI. Case Study - Implementation of a Software Sensor Based on LSTM Neural Networks in WWTP

This case study provides a comprehensive examination of incorporating ANNs as software estimators in the context of wastewater treatment, with a particular focus on predicting concentrations of ammonia in effluent.

The proposed software sensor was tested within the Simulink BSM2 model to determine the amount of $S_{NH,e}$ (ammonium) in effluent and to predict timely possible increases in its values [13].

- **Dataset and Simulation Model:** To implement this application, a representative dataset was collected from the BSM2 model, including information about key values and parameters, water quality measurements, and control variables. This dataset was used to train and test the LSTM neural network [13].
- **Training and Validation of LSTM Neural Network:** To achieve accurate predictions, the LSTM neural network was trained using machine learning algorithms on the available dataset. Subsequently, the network's performance was validated using validation data to assess the accuracy of its predictions [13].
- **Performance Evaluation of the Software Sensor:** To evaluate the performance of the implemented software sensor, its predictions were compared with the actual values measured within the WWTP simulation model, BSM2. Multiple performance indicators, such as Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2), were used to determine the accuracy and effectiveness of the software sensor [13].

Effluent and influent measurements of the treatment plants were collected over a one-year period, with a sampling interval of 15 minutes. Figure 4 illustrates the effluent prediction block structure, as well as the internal structure of the data processing block..

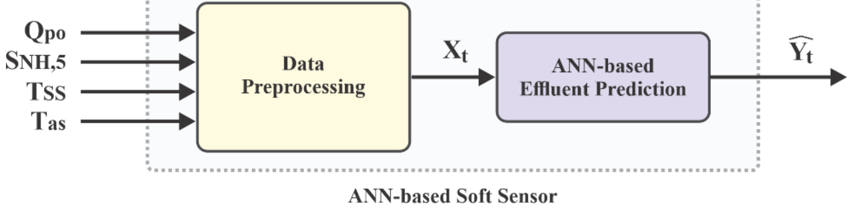


Fig. 4 *Complete structure of a software sensor based on RNA* [13]

As seen in the above figure, the system inputs include influent measurements obtained from the treatment plants. These measurements include water flows and nutrient concentrations, such as the concentration of ammonia in the fifth reservoir of the bioreactor (S_NH,5), the output flow from the first clarifier (Q_po), ambient temperature (T_as), and total suspended solids (STp). The goal of the Artificial Neural Network (ANN) is to predict the effluent concentration (ŷ). ANNs rely on a prediction methodology that involves the use of long short-term memory (LSTM) cells.

Data preprocessing techniques play a critical role in optimizing the performance and reducing the complexity of Artificial Neural Networks (ANN). Therefore, the data preprocessing approach includes three main steps, namely sliding window, data normalization, and K-Fold-based training.

The sliding window is a data preprocessing technique that involves dividing a data sequence into smaller segments or windows. It incorporates two fundamental variables, namely the window length (WL) and the prediction horizon (PH).

In this case, the WL and PH configuration were established as follows: A window length (WL) of 10 hours was selected as a suitable time interval to retain observed values at each sampling moment and include previous measurements. A Prediction Horizon (PH) of 4 hours was considered. This parameter defines the time period within which effluent concentration predictions can be provided in advance, facilitating proactive decision-making. As shown in Figure 5, the system assimilates previous data recorded for 10 hours for each new measurement. The architectural specifications of the treatment plant considered determined that the necessary retention time should be 14 hours. The sliding window technique is implemented so that with each movement of the window, a new measurement is generated, while the oldest measurement is removed according to the First-In-First-Out (FIFO) principle.

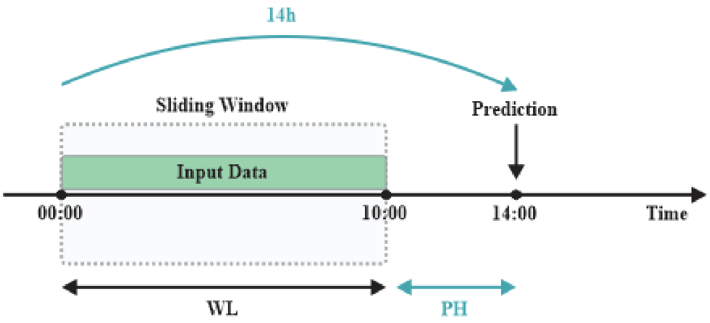


Fig. 5 *Implementation of the Sliding Window Principle* [13]

K-Fold Based Training operates on two fundamental principles: dividing the dataset into equal-sized subsets and executing training processes. In our specific case, the dataset includes influent and effluent measurements from the BSM2 model of the WWTP. The number of folds (K) was carefully chosen to allocate 70% of the complete dataset for training ANNs, while reserving 30% for testing and validation purposes. Within this 30%, 15% is designated for validation, while the remaining 15% serves as the testing subset. The goal is to obtain distinct prediction models through each training process, resulting in a total of K models. The dataset is used for each model, with subsets used for training and a dedicated subset for testing and validation. Ultimately, the model that shows superior prediction accuracy across all training processes is selected for final application.

This study uses three distinct metrics to evaluate the model's performance. These values include Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The combination of these values provides a detailed assessment of the effectiveness of the artificial neural network (ANN)-based soft sensor. The evaluation results indicate that the ANN model exhibits remarkable accuracy in its predictions (see Figure 6 and Figure 7).

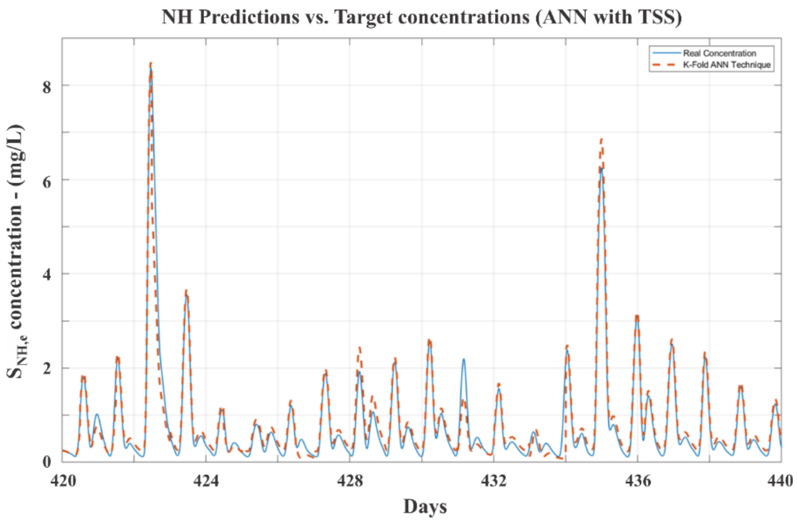


Fig.6 Implementation results of the software sensor, ANN with TSS [13]

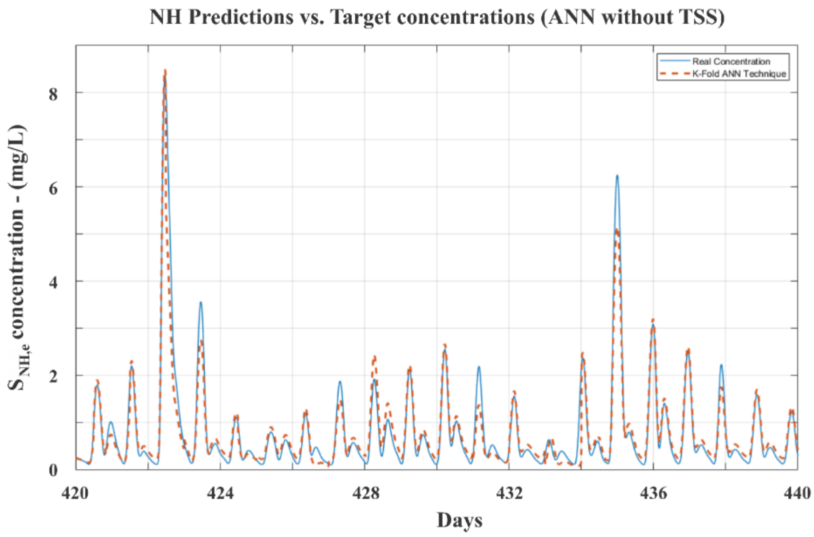


Fig.7 Implementation results of the software sensor, ANN without TSS [13]

As observed, the results obtained from the implementation of the software sensor based on LSTM neural networks in a WWTP simulation model demonstrate that this technological approach brings significant improvements in monitoring and controlling wastewater treatment processes.

Bibliography

[1] L. CONDRACHI Eng Scientific supervisor and P. Marian BARBU, “CONTRIBUTIONS REGARDING THE AUTOMATIC CONTROL OF ANAEROBIC DIGESTION PROCESSES,” 2022.

[2] “Tipuri de senzori. Senzori smart pentru proiectele tale automatizate IoT – Robofun Blog de Robotică & Electronică.” https://blog.robofun.ro/2020/04/13/tipuri-de-senzori-senzori-smart-pentru-proiectele-tale-automatizate-iot/ (accessed Oct. 30, 2023).

[3] “What is a Soft Sensor or Software Sensor? - Körber Pharma.” https://www.koerber-pharma.com/blog/what-is-a-soft-sensor-or-software-sensor (accessed Oct. 30, 2023).

[4] P. Kadlec, B. Gabrys, and S. Strandt, “Data-driven Soft Sensors in the process industry,” *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, Apr. 2009, doi: 10.1016/J.COMPCHEMENG.2008.12.012.

[5] A. Nair, A. Hykkerud, and H. Ratnaweera, “Estimating Phosphorus and COD Concentrations Using a Hybrid Soft Sensor: A Case Study in a Norwegian Municipal Wastewater Treatment Plant,” *Water 2022, Vol. 14, Page 332*, vol. 14, no. 3, p. 332, Jan. 2022, doi: 10.3390/W14030332.

[6] A. V. Luca, M. Simon-Várhelyi, N. B. Mihály, and V. M. Cristea, “Data Driven Detection of Different Dissolved Oxygen Sensor Faults for Improving Operation of the WWTP Control System,” *Process. 2021, Vol. 9, Page 1633*, vol. 9, no. 9, p. 1633, Sep. 2021, doi: 10.3390/PR9091633.

[7] M. Alvi, T. French, R. Cardell-Oliver, P. Keymer, and A. Ward, “Cost Effective Soft Sensing for Wastewater Treatment Facilities,” *IEEE Access*, vol. 10, pp. 55694–55708, 2022, doi: 10.1109/ACCESS.2022.3177201.

[8] Y. Xie *et al.*, “Enhancing Real-Time Prediction of Effluent Water Quality of Wastewater Treatment Plant Based on Improved Feedforward Neural Network Coupled with Optimization Algorithm,” *Water 2022, Vol. 14, Page 1053*, vol. 14, no. 7, p. 1053, Mar. 2022, doi: 10.3390/W14071053.

[9] H. Liu, M. Huang, and C. K. Yoo, “A fuzzy neural network-based soft sensor for modeling nutrient removal mechanism in a full-scale wastewater treatment system,” *New pub Balaban*, vol. 51, no. 31–33, pp. 6184–6193, 2013, doi: 10.1080/19443994.2013.780757.

[10] H. Xiao, B. Bai, X. Li, J. Liu, Y. Liu, and D. Huang, “Interval multiple-output soft sensors development with capacity control for wastewater treatment applications: A comparative study,” *Chemom. Intell. Lab. Syst.*, vol. 184, pp. 82–93, Jan. 2019, doi: 10.1016/J.CHEMOLAB.2018.11.007.

[11] H. Poutiainen, H. Niska, H. Heinonen-Tanski, and M. Kolehmainen, “Use of sewer on-line total solids data in wastewater treatment plant modelling,” *Water Sci. Technol.*, vol. 62, no. 4, pp. 743–750, Aug. 2010, doi: 10.2166/WST.2010.317.

[12] H. Haimi, M. Mulas, F. Corona, and R. Vahala, “Data-derived soft-sensors for biological wastewater treatment plants: An overview,” *Environ. Model. Softw.*, vol. 47, pp. 88–107, Sep. 2013, doi: 10.1016/J.ENVSOFT.2013.05.009.

[13] A. E. Ţîru, I. Vasiliev, L. Diaconu, R. Vilanova, D. Voipan, and H. Ratnaweera, “Integration of ANN for Accurate Estimation and Control in Wastewater Treatment,” *2023 IEEE 28th Int. Conf. Emerg. Technol. Fact. Autom.*, pp. 1–4, Sep. 2023, doi: 10.1109/ETFA54631.2023.10275569.

Practical activity 1

Data files collected from sensors. Uploading raw data

Sensor software simulation

Context

Implementing sensor software involves several essential aspects such as:

- Data collection,
- Data inspection,
- Selecting historical data,
- Data preprocessing,
- Selecting a mathematical model,
- Training and validating the model,
- Integrating the sensor software
- Monitoring and maintaining the sensor software

Internet of Things (IoT) devices constantly generate data in our environment. Python is a powerful tool for analyzing this data. In this lab, you will explore various types and Python data modules to learn how to read, interpret, and transform data from one file to another.

Objectives

In this first module, you will develop a function that allows you to upload data generated by a sensor and stored in separate files. This data is expressed in various numerical forms and is recorded in the standard CSV format.

Using the Python language for data analysis and processing

1.1 Introductory Concepts

File manipulation in Python is an essential operation when working with data stored in files. The 'with' block is a recommended way to open and work with files in Python because it ensures proper resource management and avoids memory leaks.

1. Opening a file:

To open a file in Python, use the *`open()`* function[1]. This function takes two main arguments: the file name and the mode of opening (read, write, etc.).

Example:

```
with open('nume_fisier.txt', 'r') as file:
```

2.Block *`with`*:

The *with* block is used to create a context in which resources (such as files) are managed automatically. Upon entering the 'with' block, the file is opened, and upon exiting the block, the file is automatically closed, regardless of whether the program execution was normal or an exception occurred..

3.Working with the file:

Inside the *with* block, you can work with the file using the 'file' variable, which is an open file object. You can read, write, or modify the content of the file using methods and operations specific to the file object.

Example:

```
with open('nume_fisier.txt', 'r') as file:
    data = file.read()
    print(data)
```

4.Closing the file:

The *with* block ensures the automatic closing of the file when the execution reaches its end. There is no longer a need to manually close the file using file.close(). Closing the file is important for resource release and to avoid memory leaks.

1.2 Exercises

1.The dataset used in this lab is scattered in a file named SENSOR_1.CSV, located in the datasets folder. This data represents information from a device equipped with various sensors. The information was randomly collected over a period of several days and includes measurements related to temperature, humidity, energy consumption, and airborne particle concentration in a specific area. Data collection took place over a 24-hour period.

1. The purpose of the exercise is to create a Python function that loads this data from CSV files. Follow the steps below to solve the exercise: Create a function named load_sensor_data that takes a single parameter, directory_path, representing the path to the directory containing CSV files with sensor data.
2. Initialize an empty list named sensor_data to store the data.
3. Get the list of CSV files in the specified directory and iterate through each file.
4. Inside the loop for each file, use a with block to open the CSV file.
5. Use *csv.DictReader* to read the data from the file and add each record to the sensor_data list.
6. Return the sensor_data list with the loaded data.
7. Usage example: Call the load_sensor_data function with the path to the directory containing CSV files and display the first few records to verify correct functionality.

Note: Make sure to correctly specify the path to the directory with CSV files for accurate results.

2.Develop a software sensor that can monitor the Chemical Oxygen Demand (COD) level in a wastewater treatment plant.

1. The software sensor should generate random values for the COD level between 10 and 1000 mg/L. It should be user-friendly and provide simulated data in an easily readable format.

2. It should display the simulated COD level every 0.1 seconds.
3. The program should run in an infinite loop to simulate periodic measurements.

Bibliography

[1] “3.8.17 Documentation.” https://docs.python.org/3.8/ (accessed June 07, 2023).

Big Data in wastewater treatment processes

1. Big Data – Concepts

In today's digital era, the volumes of data generated and collected each day are enormous, giving rise to the concept of "Big Data." Big Data refers to massive collections of data, both structured and unstructured, that cannot be efficiently managed and analyzed using traditional tools and technologies. This phenomenon has fundamentally changed how we organize, store, analyze, and understand information in the modern world.

Big Data is fueled by a variety of sources, including Internet of Things (IoT) devices, social media, monitoring systems, advanced sensor technologies, and many more. This data comes in the form of text, images, videos, audio messages, and more, transforming how we manage and utilize information across various domains, from business and medicine to scientific research and governance.

There are four key characteristics that define Big Data, known by the acronym "V": Volume, Velocity, Variety, and Value:

a) **Volume:** Big Data is characterized by massive quantities of data. These data volumes are typically much larger than what traditional database systems could handle. Therefore, storing and managing these enormous data volumes requires specialized infrastructure and technologies.

b) **Velocity**: Data is generated and proliferates at an impressive speed. Information can be updated in real-time, and the ability to respond quickly to changes and events is crucial. This characteristic is often associated with IoT, where data is generated at very short intervals.

c) **Variety**: Big Data can have significant variety in terms of formats and sources. Data can be in the form of text, images, video, geospatial data, and more. Integrating and analyzing this diverse data poses a significant challenge.

d) **Value**: The primary goal of Big Data is to extract value from these large and diverse volumes of data. Through advanced analysis, patterns, trends, and insights can be discovered, supporting decision-making, innovation, and efficiency in various fields.

Big Data presents significant opportunities but also comes with major challenges. Managing data on such a large scale requires significant investments in infrastructure, technology, and expertise. Issues related to security and privacy are also highly important, as the more data becomes available, the higher the risk of exposing sensitive information.

Despite the challenges, Big Data is a powerful driver of innovation and process improvement across all fields. With advanced tools and technologies, Big Data provides us with the opportunity to better understand the world around us, develop more efficient solutions, and make more accurate predictions. It is a continually evolving field that promises to bring substantial benefits to society and the global economy.

2. Big Data Technologies

Big Data technologies represent a set of tools, techniques, and technologies used for managing, storing, processing, and analyzing massive volumes of data. These technologies are crucial in today's data-driven world as they enable organizations to extract value from vast amounts of information. Here is a theoretical overview of the main Big Data technologies:

a) Hadoop

Hadoop Distributed File System (HDFS): This is a distributed file system that allows data storage across a cluster. Data is divided into blocks and replicated across multiple nodes for resilience.
MapReduce: A programming model and framework for parallel processing of large data. It is used for operations like filtering and aggregation on large datasets, such as sorting and filtering.

b) Apache Spark

In-Memory Processing: Spark uses memory instead of disk for processing, making it much faster than MapReduce.
Versatile APIs: Provides APIs for Python, Scala, and Java, making it easier for developers to create complex data analysis applications.
Support for Stream Analysis: Spark Streaming allows processing and analyzing data in real-time, useful in areas like log analysis or fraud detection.

c) NoSQL Databases

MongoDB: A document-oriented database used for storing unstructured or semi-structured data, such as JSON documents.
Cassandra: A distributed and scalable database, ideal for applications with high performance and scalability requirements.
Redis: A memory database used for caching and real-time processing.

d) Columnar Databases

Amazon Redshift: A columnar data warehouse ideal for data analysis in the cloud.
Google BigQuery: A cloud-based real-time data analysis database, allowing analysis of data in real-time.

e) Data Warehouses

Teradata: An enterprise solution for data management and business intelligence analysis.
Snowflake: A cloud-based data warehouse, scalable and efficient in data management.

f) Machine Learning and Artificial Intelligence

Big Data is essential for training machine learning models on large and diverse datasets.
TensorFlow, scikit-learn, and PyTorch are popular libraries used for developing ML models.

g) Stream Processing

Apache Kafka: A distributed messaging system that allows processing data streams in real-time.
Apache Flink and Apache Storm: Frameworks for real-time stream processing.

h) Cloud Computing

Cloud services offer scalable resources for data storage and analysis.

AWS, Microsoft Azure, and Google Cloud Platform provide Big Data and Machine Learning services.

i) Big Data Ecosystems

Ecosystems provide a suite of tools and services for comprehensive Big Data management. Cloudera, Hortonworks, and MapR are notable examples.

j) Data Integration Tools

ETL tools like Apache Nifi, Talend, and Informatica enable extraction, transformation, and loading of data from diverse sources into the Big Data environment.
These technologies are used in various fields, from financial and medical analysis to log analysis and supply chain optimization. In each domain, Big Data technologies are essential for extracting valuable insights from existing data and making more informed decisions. However, resource requirements, costs, and security requirements must be considered to successfully implement Big Data solutions.

3. Using Big Data techniques in wastewater treatment plants

Water is an essential resource for life on Earth, and the proper management and treatment of wastewater are major concerns in modern society. With the increasing global population and ongoing industrialization, the management and treatment of wastewater are becoming increasingly complex. In this context, Big Data proves to be an innovative tool with a significant role in wastewater treatment processes, bringing a series of benefits for the efficiency and quality of these processes.

One of the most crucial aspects of utilizing Big Data in wastewater treatment is data collection. Advanced sensors and monitoring tools gather essential data on water quality, pollution levels, wastewater quantities, and more. This data forms the basis for understanding the current state of the wastewater treatment system. By collecting these detailed data sets, operators have essential information for making correct real-time decisions. Another critical aspect of Big Data is data analysis. The massive amounts of collected data can be analyzed to identify trends and similarities that can be used to enhance wastewater treatment processes. This provides an opportunity to develop predictive models and anticipate changes in water quality or pollution levels. Through a deeper understanding of the data, more efficient solutions for wastewater treatment can be identified.

Optimizing processes is another benefit brought by Big Data in wastewater treatment. Data analysis can reveal ways to adjust processes, such as chemical dosages, water flow control, and reducing energy consumption. This not only contributes to resource savings but also reduces the impact on the environment. Big Data is also useful in detecting abnormal events. The collected data can be used to identify water pollution incidents or other unexpected issues. Operators can receive real-time alerts about these situations, allowing them to react quickly and minimize the impact on the environment.

Forecasting maintenance needs is another essential aspect. Through the analysis of Big Data, maintenance needs for equipment and infrastructure used in wastewater treatment can be anticipated. This helps reduce downtime and maintenance costs while ensuring the continuous operation of wastewater treatment systems.

In addition to all these advantages, Big Data contributes to cost reduction in wastewater treatment operations through process optimization and efficient resource management. This means a more efficient use of financial and material resources, positively impacting the sustainability and budget of wastewater treatment organizations.

Another important aspect is regulatory compliance. Big Data can be used to generate accurate reports and demonstrate compliance with environmental standards and government regulations regarding water quality. This ensures that wastewater treatment systems operate in accordance with standards and that the released water meets all quality requirements. Additionally, Big Data is essential in real-time water quality monitoring. This ensures that the treated water is safe for the environment and human consumption. The collected data allows for the rapid detection of any issues in water quality, and operators can take immediate action to address the situation. Big Data represents a revolution in wastewater treatment, bringing a series of significant benefits for the efficiency and quality of these processes. Data collection, analysis, process optimization, abnormal event detection, maintenance needs forecasting, cost reduction, regulatory compliance, and water quality monitoring are all essential aspects that Big Data brings to this vital field. It provides operators with the opportunity to make more informed decisions and respond more efficiently to wastewater treatment needs, contributing to environmental protection and ensuring adequate drinking water supply for society.

Examples of Big Data applications:

1) **Sensor Networks**: Sensor technology forms the basis for Big Data applications in wastewater treatment plants (WWTP). These sensors are strategically placed throughout the treatment plant to monitor various parameters such as water quality, flow rates, temperature, and more. They continuously collect data and transmit it to a central system for analysis.

2) **Data Storage and Management**: Robust data storage and management systems are necessary to handle the large volumes of generated data. This typically involves the use of distributed databases and data warehouses capable of efficiently storing and retrieving data.

3) **Real-Time Monitoring**: Big Data technologies enable real-time monitoring of WWTP processes. Operators can access real-time data streams and receive alerts regarding deviations or anomalies, allowing for immediate responses to optimize treatment processes.

4) **Data Analysis and Machine Learning**: Advanced data analysis techniques and machine learning are used to extract valuable insights from collected data. These technologies can help identify patterns, prevent equipment failures, optimize chemical dosages, and enhance overall station performance.

5) **Predictive Maintenance**: Predictive maintenance is crucial for WWTP. Using historical data and machine learning models, maintenance schedules can be optimized to reduce downtime and minimize equipment failures. This ensures the efficient and cost-effective operation of the treatment plant.

6) **Data Visualization**: Data visualization tools are used to create user-friendly dashboards and reports for station operators and decision-makers. Visual representations of data help quickly identify trends and issues.

7) **Remote Monitoring and Control**: Big Data technologies enable remote monitoring and control of WWTP processes. This is particularly valuable for large treatment facilities where immediate on-site responses may not be possible.

8) **Integration with Smart Technologies**: Integration with other smart technologies, such as Internet of Things (IoT) devices, can further enhance WWTP capabilities. IoT sensors can provide additional data sources for monitoring and control.

4. Big Data Analytics with Python

Big Data Analytics is a field that focuses on processing and analyzing massive amounts of data to gain valuable insights. **Apache Spark** is an open-source data processing framework designed to handle large volumes of data and provide real-time performance. Spark employs concepts such as **RDDs** (Resilient Distributed Datasets) and transformation and action operations to perform data analysis on a distributed computing cluster. RDDs are fundamental data structures in Apache Spark. They are distributed datasets that can be parallelized and are fault tolerant. RDDs can be created from data from external sources and processed in a distributed manner across the nodes of the cluster.

PySpark is a Python library that enables working with massive data using Apache Spark, a distributed data processing framework. Here are some fundamental theoretical concepts in Big Data Analytics with PySpark. Using PySpark in the analysis of data from wastewater treatment plants (WWTP) can be extremely beneficial for optimizing wastewater treatment processes, improving efficiency, and reducing costs. Examples of how PySpark can be applied in this context:

1) **Data Collection and Preparation**: PySpark can be used to collect and process data from sensors in WWTP, including water quality, pollution levels, water flow, and more. Data can be imported from various sources, including CSV files, databases, or directly from sensors.
2) **Data Storage and Management**: Data can be stored in a distributed format, such as RDDs or DataFrames, to enable distributed processing with Spark. PySpark can be used to integrate data from various sources and perform transformations, such as data cleaning or extracting relevant information.
3) **Data Analysis and Processing**: With PySpark, you can perform advanced data analysis in WWTP, including identifying trends, correlations, and patterns in the collected data. For example, you can analyze how pollution levels vary based on seasons or weather conditions.
4) **Process Optimization**: PySpark can be used to optimize processes in WWTP. For instance, machine learning algorithms can be applied to optimize chemical dosages or predict equipment maintenance needs.
5) **Detection of Anomalous Events**: PySpark can help detect abnormal events or pollution incidents in WWTP. Anomaly detection models can alert operators in real-time in case of issues, enabling quick responses.
6) **Data Visualization**: PySpark provides facilities to create relevant and easy-to-understand data visualizations. These visualizations can help efficiently communicate information to WWTP staff and decision-makers.
7) **Machine Learning in WWTP**: PySpark includes MLlib, a module that can be used to build and train machine learning models in WWTP. These models can be used to make predictions and support decisions related to wastewater treatment.
8) **Real-time Data Streaming**: Using Spark Streaming, a component of Apache Spark, you can analyze data from WWTP in real-time. This is useful for real-time monitoring of water quality and detecting unexpected events.
9) **Integration with Intelligent Technologies**: Integration with intelligent technologies, such as IoT devices for real-time data collection, can complement WWTP data analysis and enable faster and more efficient decision-making.

The use of PySpark in conjunction with data from WWTP allows for the improvement of wastewater treatment processes, the reduction of operating costs, and ensuring compliance with environmental protection regulations. This can contribute to the protection of the environment and ensure efficient wastewater treatment.

The official PySpark documentation is an excellent resource to learn more about this data processing framework and to find detailed information about its functionalities and APIs. When working with PySpark, the official documentation is the best resource to find answers to your questions and learn how to efficiently use this powerful data processing framework:

https://spark.apache.org/docs/latest/api/python/getting_started/index.html

4.1 Using Pyspark in Google Colab

Google Colab (Colaboratory) is a free Jupyter Notebook platform provided by Google, running in the cloud, allowing you to work with Python and other libraries. Additionally, you can use PySpark in Google Colab for Big Data analysis. Here are the basic steps to start working with PySpark in Google Colab.

PySpark, Apache Spark's framework for programming in the Python language, is designed to process and analyze large volumes of data efficiently and scalably. The basic elements of PySpark are described below. These elements help you manage, process, and analyze large volumes of data efficiently and in a distributed manner. PySpark is powerful, versatile, and used in a variety of fields, from data analysis to machine learning and real-time data analysis.

SparkSession: This is the main entry point in PySpark and is used to create a Spark session. Through the Spark session, you can configure settings, load data, create DataFrames and RDDs, and interact with Spark clusters.

DataFrame: DataFrame is a tabular data structure, similar to a table in a relational database. It is a fundamental concept in PySpark and is used to manipulate data efficiently. Data can be loaded into a DataFrame from various sources, such as CSV files, databases, or other structured data sources.

Resilient Distributed Dataset (RDD): RDD is another fundamental concept in Apache Spark and represents a distributed and immutable collection of data. RDDs are used for distributed data processing on Spark clusters and are useful in more advanced scenarios.

Transformations and Actions: PySpark provides a series of transformations and actions on DataFrames and RDDs. Transformations are lazy operations that transform an existing DataFrame into a new one, while actions effectively trigger operations and return results. Examples of transformations include select(), filter(), groupBy(), while examples of actions include show(), count(), saveAsTable().

Partitions: Data in a DataFrame or RDD is divided into multiple partitions to enable distributed processing across multiple nodes. Partitions are basic units for distribution and parallel processing of data.

Spark Clusters: To use PySpark efficiently, a cluster infrastructure is required. A Spark cluster consists of a group of nodes working together to process and analyze data. Clusters can be managed using solutions such as Apache Hadoop, YARN, or in cloud environments like AWS EMR or Google Dataproc.

Storage and Parallelism: PySpark offers options for distributed data storage and ensuring parallelism in data processing. This ensures that data can be processed efficiently on the Spark cluster.

Python Libraries: You can use Python libraries such as NumPy, Pandas, Matplotlib, and others for data analysis and visualization within the PySpark environment.

Optimization and Tuning: Apache Spark provides advanced optimization capabilities, such as in-memory storage and efficient operation scheduling, to accelerate data processing.

Machine Learning with MLlib: Spark MLlib is a library for machine learning and provides machine learning algorithms for classification, regression, clustering, and more.

Below are the steps for installing and using PySpark in Google Colab:

Step 1: Google Colab already comes with Python installed, but you need to install PySpark and set up a Spark session. Use the following code in a code cell in Colab:

```
# Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
# Install Spark (modify version number if neccessary)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-
bin-hadoop3.2.tgz

# Unzip the Spark file in the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

# Set the Spark directory path to your system's environment variables
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

# Install findspark using pip
!pip install -q findspark
```

Step 2: Set the environment variables for Java and Spark. You will also need to adjust the Spark version in the specified path to match the installed version:

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"
```

Step 3: Add folders Spark and PySpark to the system path:

```
import findspark
findspark.init()
```

Step 4: Import PySpark and create a SparkSession. This example demonstrates how to configure Spark to run locally on Google Colab.

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("example").getOrCreate()
```

Step 5: Working with PySpark. After setting up a Spark session, you can start working with PySpark in other cells of the notebook. You can create RDDs or DataFrames and perform data analysis operations using the PySpark library. You can load data into Google Colab from various sources, such as Google Drive, Dropbox, or directly from a URL. Once you have loaded the data, you can use PySpark to process it.

```
# Create a DataFrame
data = [("Alice", 34), ("Bob", 45), ("Catherine", 29)]
columns = ["Name", "Age"]
df = spark.createDataFrame(data, columns)
# Afisati DataFrame
df.show()

# Perform a simple analysis operation
df.select("Name", "Age").filter(df.Age > 30).show()
```

Step 6: Exporting Results. After you have conducted the data analysis and obtained the desired results, you can export the results to files or share the notebook with colleagues or others.

Step 7: When you're done, don't forget to stop the Spark session to release resources:

```
spark.stop()
```

Wastewater data analysis may involve examining and visualizing data related to the quality and quantity of wastewater. We will use the Python language and some common data science libraries, such as Pandas, Matplotlib, and Seaborn, to explore and visualize wastewater data. Let's assume you have a dataset containing information about the quality and quantity of wastewater over time. Here's a simplified example of how you can approach the analysis:

Firstly, you need to load the wastewater data from a CSV or Excel file into a Pandas DataFrame. For this example, let's assume you have a CSV file named "date_apa_uzata.csv".

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("AnalizaApaUzata").getOrCreate()

# Load the Data into a DataFrame PySpark
data_apa_uzata_df = spark.read.csv("date_apa_uzata.csv", header=True,
inferSchema=True)
```

Explore the dataset to understand its structure and content. You can check the first few rows of the dataset, summary statistics, and data types.

```
# Display the first few rows of the DataFrame
data_apa_uzata_df.show()

# Display summary statistics
```

```
data_apa_uzata_df.describe().show()

# Check data types
data_apa_uzata_df.printSchema()
plt.ylabel("Cantitate (m³/zi)")
plt.show()
```

Visualize the data using PySpark with Matplotlib and Seaborn:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Convert the PySpark DataFrame to a Pandas DataFrame for visualization

data_apa_uzata_pandas_df = data_apa_uzata_df.toPandas()

# Create a time series plot
plt.figure(figsize=(12, 6))
sns.lineplot(x="Data", y="CantitateApaUzata",
data=data_apa_uzata_pandas_df)
plt.title("Cantitatea de Apă Uzată în Timp")
plt.xlabel("Data")
plt.ylabel("Cantitate (m³/zi)")
plt.show()

# Create a histogram for wastewater quality
plt.figure(figsize=(8, 6))
sns.histplot(data_apa_uzata_pandas_df["CalitateApaUzata"], bins=20,
kde=True)
plt.title("Distribuția Calității Apei Uzate")
plt.xlabel("Calitate")
plt.ylabel("Contor")
plt.show()
```

Perform data analysis and gain insights using PySpark capabilities for data transformation, filtering, and aggregation. Make recommendations based on the data analysis. Don't forget to stop the PySpark session when you're finished:

```
spark.stop()
```

This example allows you to perform wastewater data analysis using PySpark for distributed data processing. A real-world wastewater data analysis case may involve more extensive data preprocessing, modeling, and domain-specific knowledge. Additionally, you can enhance this example by incorporating machine learning models for predictive analysis or anomaly detection if your dataset allows for such analyses.

4.2. Examples of data processing operations.

With PySpark, you can perform a variety of data processing and analysis operations on the data collected from wastewater treatment plants. Here are some typical operations you can perform:

- Data Preprocessing: Cleaning and transforming data to remove missing or incorrect data.
- Standardizing and normalizing data if collected from diverse sources or has different units.
- Data Filtering and Selection: Filtering data to extract subsets relevant for analysis.
- Selecting data based on specific criteria, such as time intervals or specific water quality parameters.
- Data Aggregation: Calculating basic statistics such as mean, standard deviation, minimum, and maximum for various water quality parameters.
- Aggregating data based on specific time intervals or locations to gain global or regional perspectives on wastewater quality.
- Time Series Analysis: Conducting time series analysis to identify trends, seasonal patterns, and variations in wastewater parameters.
- Detecting anomalies in time series data that could indicate issues in the functioning of the treatment plant.
- Correlation Analysis: Assessing correlations between different wastewater parameters to understand their relationships.
- Determining the impact of factors on wastewater quality.
- Regression Analysis: Using regression models to predict future values of wastewater parameters based on historical data.
- Visualization and Reporting: Creating charts, diagrams, and reports to communicate analysis results to stakeholders or for decision-making.
- Cost and Efficiency Analysis: Evaluating the costs and efficiency of operations at the wastewater treatment plant.
- Identifying potential improvements to reduce costs or optimize processes.
- Machine Learning and Artificial Intelligence: Implementing machine learning models to make predictions, for example, anticipating issues before they occur.
- Implementing AI algorithms for optimizing control and monitoring processes.

Example 1: Let's consider a simplified example in Python, using PySpark, to perform some common operations on wastewater data from a wastewater treatment plant. In this example, we assume that you have a PySpark environment configured and that you have a dataset in a CSV file named "date_apa_uzata.csv".

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, avg, stddev, max

# Create a Spark session
spark = SparkSession.builder.appName("ExempluApaUzata").getOrCreate()

# Load the wastewater data into a DataFrame PySpark
data_apa_uzata_df = spark.read.csv("date_apa_uzata.csv", header=True,
inferSchema=True)

# Display the first few rows of the DataFrame-ului
data_apa_uzata_df.show()
```

```
# Calculate and display the mean and standard deviation of the column
"CantitateApaUzata"
medie_cantitate_apa =
data_apa_uzata_df.select(avg(col("CantitateApaUzata"))).first()[0]
deviatie_standard =
data_apa_uzata_df.select(stddev(col("CantitateApaUzata"))).first()[0]
print(f"Medie Cantitate Apă Uzată: {medie_cantitate_apa}")
print(f"Deviație Standard a Cantității de Apă Uzată:
{deviatie_standard}")

# Find the date when the maximum water quality was recorded.
data_max_calitate =
data_apa_uzata_df.select("Data").filter(data_apa_uzata_df.CalitateApaUz
ata == max(data_apa_uzata_df.CalitateApaUzata)).first()
print(f"Data cu Calitatea Maximă a Apei Uzate:
{data_max_calitate.Data}")

# Group the data by month and calculate the average of the Wastewater
Quantity for each month.
data_apa_uzata_df = data_apa_uzata_df.withColumn("Luna",
data_apa_uzata_df["Data"].substr(1, 7))
medie_lunara_cantitate_apa =
data_apa_uzata_df.groupBy("Luna").agg(avg("CantitateApaUzata").alias("M
edieCantitateApa"))
medie_lunara_cantitate_apa.show()

# Stop the Spark session
spark.stop()
```

In the example above, the steps are as follows:

- Loading the wastewater data into a PySpark DataFrame.
- Calculating and displaying the mean and standard deviation of the "WastewaterQuantity" column.
- Finding the date when the maximum "WaterQuality" was recorded.
- Grouping the data by month and calculating the average "WastewaterQuantity" for each month.

This is a simple example illustrating how you can use PySpark to perform operations on wastewater data. You can extend this example with more complex analyses, visualization, or machine learning techniques based on specific requirements.

Example 2: In this example, we focus on data transformation and aggregation.

```
# Import PySpark modules
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, when

# Create a Spark session
```

```
spark = SparkSession.builder.appName("ExempluApaUzata2").getOrCreate()

# Load the wastewater data into a DataFrame PySpark (replace
"date_apa_uzata.csv" with your data file)
data_apa_uzata_df = spark.read.csv("date_apa_uzata.csv", header=True,
inferSchema=True)

# Create a new column "QualityCategory" based on "CalitateApaUzata"
data_apa_uzata_df = data_apa_uzata_df.withColumn("QualityCategory",
when(col("CalitateApaUzata") >= 80, "Calitate Înaltă")
                    .when(col("CalitateApaUzata") >= 50, "Calitate
Medie")
                    .otherwise("Calitate Scăzută"))

# Compute the total quantity of wastewater in a month
data_apa_uzata_df = data_apa_uzata_df.withColumn("Luna",
data_apa_uzata_df["Data"].substr(1, 7))
cantitate_totala_lunara =
data_apa_uzata_df.groupBy("Luna").sum("CantitateApaUzata")

# Identify the month with the highest quantity of water
luna_maxima_cantitate =
cantitate_totala_lunara.orderBy("sum(CantitateApaUzata)",
ascending=False).first()
print(f"Luna cu Cea Mai Mare Cantitate Totală:
{luna_maxima_cantitate['Luna']}")

# Save the transformed DataFrame-ului in a new CSV file
data_apa_uzata_df.write.csv("date_apa_uzata_transformate.csv",
header=True)

# Stop the Spark session
spark.stop()
```

In the example above, the steps are:

- Creating a new column "QualityCategory" based on the "WaterQuality" column, categorizing the quality as "High Quality," "Medium Quality," or "Low Quality."
- Calculating the total wastewater quantity per month and identifying the month with the highest total quantity.
- Saving the transformed DataFrame to a new CSV file, which includes the "QualityCategory" column.

This example demonstrates how to perform data transformations, aggregations, and save the results in a new file using PySpark. You can adapt and extend these operations to suit your specific wastewater data analysis needs.

Example 3: This example illustrates how you can use PySpark for more advanced wastewater data analysis tasks, including data integration, machine learning, and predictive modeling. Make sure to replace "date_apa_uzata.csv" and "date_meteorologice.csv" with the actual file names of your datasets.

```
# Import PySpark modules
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression

# Create a Spark session
spark = SparkSession.builder.appName("ExempluApaUzata3").getOrCreate()

# Load the wastewater data into a DataFrame PySpark
data_apa_uzata_df = spark.read.csv("date_apa_uzata.csv", header=True,
inferSchema=True)

# Load the meteorological data into a DataFrame PySpark (replace it
with your file of meteorological data)
date_meteorologice_df = spark.read.csv("date_meteorologice.csv",
header=True, inferSchema=True)

# Merging the two DataFrames using a common column (for example "Data")
data_combinata_df = data_apa_uzata_df.join(date_meteorologice_df,
"Data", "inner")

# Performing data preparation, including feature vectorization
coloane_caracteristici = ["CalitateApaUzata", "Temperatura",
"Precipitații"]
asamblor = VectorAssembler(inputCols=coloane_caracteristici,
outputCol="caracteristici")
date_pregatite = asamblor.transform(data_combinata_df)

# Define a Linear Regression model
rl = LinearRegression(featuresCol="caracteristici",
labelCol="CantitateApaUzata")

# Split the data into training and testing sets
date_antrenament, date_testare = date_pregatite.randomSplit([0.7, 0.3])

# Train the linear regression model
model_rl = rl.fit(date_antrenament)

# Making predictions on the test data.
previziuni = model_rl.transform(date_testare)

# Evaluate the model performance
from pyspark.ml.evaluation import RegressionEvaluator
evaluator = RegressionEvaluator(labelCol="CantitateApaUzata",
predictionCol="previziune", metricName="rmse")
rmse = evaluator.evaluate(previziuni)
```

```
print(f"Eroare Medie Patratică (RMSE): {rmse}")

# Stop the Spark session
spark.stop()
```

In the example above, the steps are:

- Loading two datasets, one containing wastewater data and the other containing meteorological data.
- Concatenating these datasets using a common column, in this case, the "Date" column.
- Performing data preparation, including feature vectorization, to prepare the data for machine learning.
- Defining a Linear Regression model to train on the combined dataset.
- Splitting the data into training and testing sets and evaluating the model's performance using Root Mean Squared Error (RMSE).

Bibliography

1. Al-Jasser, A. O., & Ghani, M. (2015). Big Data and Water: A Review of the State-of-the-Art and Future Opportunities. Journal of Hydroinformatics, 17(1), 16-32.
2. Palani, S., Goyal, R., & Bhatt, V. S. (2019). Big Data Analytics for Efficient Wastewater Treatment in Smart Cities. Journal of Water Process Engineering, 30, 100622.
3. Kaushal, R. K., & Patel, P. (2016). Data-Driven Modeling and Optimization of Wastewater Treatment Plants. Water Research, 88, 351-366.
4. Qasim, S. R., Hu, Y., & Chiang, P. C. (2016). Wastewater Treatment Plants: Planning, Design, and Operation. CRC Press.
5. Chen, Y., & Wang, X. (2019). Big Data Analytics in Water Resources Engineering: A Survey. CRC Press.
6. De Moura, L. J., Silva, F. M., & Luvizotto, E. (2019). Data Analytics and PySpark for Enhancing Wastewater Treatment Plants Efficiency. Chemical Engineering Transactions, 74, 1813-1818.
7. Fernandez, M. S., Torrens, R., & Ayesa, E. (2020). Leveraging PySpark for Real-Time Data Analysis and Decision Support in Wastewater Treatment Plants. Procedia Computer Science, 173, 123-130.
8. Sun, Z., et al. (2019). Real-Time Monitoring of a Municipal Wastewater Treatment Plant Using PySpark for Data Analysis. Water Research, 155, 143-153.
9. Smith, R., & Patel, A. (2020). PySpark-Based Data Integration and Analysis for Enhanced WWTP Efficiency. Journal of Environmental Engineering, 146(5), 04020010.
10. Chen, X., et al. (2018). Applying PySpark for Predictive Maintenance in Wastewater Treatment Facilities. IFAC-PapersOnLine, 51(12), 1445-1450.
11. Global Water Intelligence. (Accessed November 2, 2023). https://www.globalwaterintel.com/
12. Water Online. (Accessed November 2, 2023). https://www.wateronline.com/
13. International Water Association. (Accessed November 2, 2023). https://www.iwa-network.org/

Laboratory – Application 1

Objectives:

- Understanding big data analysis techniques applicable to experimental datasets collected from wastewater treatment stations.
- Mastery of specific terms and understanding the importance of online monitoring as a crucial aspect of digitizing wastewater treatment plants.
- Data integration, machine learning, and predictive modeling.

Introduction

In the digital era, data analysis has become a crucial tool for optimizing processes in various fields, including the management of wastewater resources. This laboratory report focuses on the use of PySpark, a data processing framework in the Python language, to analyze data related to wastewater quality and associated meteorological factors. The goal is to develop a linear regression model that can predict the quantity of wastewater based on the observed variables.

Wastewater Quality: Wastewater quality data can include parameters such as pollutant concentrations (e.g., BOD, COD, nitrogen, phosphorus, heavy metals), pH indicators, turbidity, water temperature, and other relevant parameters. This data can be collected from wastewater treatment plants or specialized laboratories.

Meteorological Data: Meteorological data may include information such as air temperature, precipitation, humidity, wind direction and speed, atmospheric pressure, and others. These data can be collected from local meteorological stations or official meteorological sources.

Step 1: Open Google Colab (https://colab.research.google.com/) **and follow the installation steps:**

```
# Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Install Spark (modify version number if neccessary)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

# Unzip the Spark file in the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

# Set the Spark directory path to your system's environment variables.
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"
```

```
# Install findspark using pip
!pip install -q findspark
```

Step 2: Configure the Spark session.

We begin by configuring a Spark session using PySpark. This session will enable us to work with big data and leverage powerful analytical functionalities.

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("ExempluAplicatia1").getOrCreate()
```

Step 3: Upload de Data

We load wastewater quality data from a CSV file and meteorological data from another CSV file. These data will be stored in PySpark DataFrames.

```
# Load wastewater data into a DataFrame PySpark
data_apa_uzata_df = spark.read.csv("date_apa_uzata.csv", header=True,
inferSchema=True)

# Load meteorological data into a PySpark DataFrame (replace with your
meteorological data file).
date_meteorologice_df = spark.read.csv("date_meteorologice.csv",
header=True, inferSchema=True)
```

Step 4: Concatenate the data.

Combine the two DataFrames using a common column, such as "Date." This will enable us to analyze the data from both sources appropriately.

```
# Join the two DataFrames using a common column ("Date") using the "join"
method.
data_combinata_df = data_apa_uzata_df.join(date_meteorologice_df, "Data",
"inner")
```

Step 5: Prepare the Data

We perform data preparation, including feature vectorization. In this case, the columns of interest include "CalitateApaUzata", "Temperatura" and "Precipitații".

```
# We define the feature columns
coloane_caracteristici = ["CalitateApaUzata", "Temperatura",
"Precipitații"]

# We create an assembler for feature vectorization
asamblor = VectorAssembler(inputCols=coloane_caracteristici,
outputCol="caracteristici")

# We apply the assembler to transform the data
```

```
date_pregatite = asamblor.transform(data_combinata_df)
```

Step 6: Implement the Linear Regression model.

We define a linear regression model that will be trained on the prepared data. This model will help predict the quantity of wastewater based on the observed features.

```
from pyspark.ml.regression import LinearRegression

# We define the Linear Regression model.
rl = LinearRegression(featuresCol="caracteristici",
labelCol="CantitateApaUzata")
```

Step 7: Split the data and train the model.

We split the data into training and testing sets and then train the linear regression model.

```
# We split the data into training and testing sets.
date_antrenament, date_testare = date_pregatite.randomSplit([0.7, 0.3])

# We train the Linear Regression model.
model_rl = rl.fit(date_antrenament)
```

Step 8: Make predictions and evaluate performance.

We use the trained model to make predictions on the test data and then evaluate the model's performance using the RMSE (Root Mean Squared Error) metric.

```
# Evaluate the model's performance.
from pyspark.ml.evaluation import RegressionEvaluator
evaluator = RegressionEvaluator(labelCol="CantitateApaUzata",
predictionCol="previziune", metricName="rmse")
rmse = evaluator.evaluate(previziuni)
print(f"Rădăcina Eroare Medie Patratică (RMSE): {rmse}")
```

```
# Stop the Spark session
spark.stop()
```

Step 9: Interpret the results and explain the operations that have been performed.

Laboratory – Application 2

Objectives:

- Understanding big data analysis techniques applicable to experimental datasets collected from wastewater treatment stations.
- Mastery of specific terms and understanding the importance of online monitoring as a crucial aspect of digitizing wastewater treatment plants.
- Data integration, machine learning, and predictive modeling.

Introduction

In this lab, we will use PySpark to analyze data related to wastewater quality. The goal is to develop a classification model to determine whether wastewater is of "Good Quality" or "Poor Quality" based on observed features.

An example source for a dataset related to wastewater quality and associated features is:

UCI Machine Learning Repository - Water Quality Data Set:

This dataset contains information about wastewater quality and water characteristics such as temperature, pH, conductivity, and others. You can download the dataset from the UCI Machine Learning Repository using the following link: https://www.kaggle.com/datasets/adityakadiwal/water-potability

Step 1: Open Google Colab (https://colab.research.google.com/) **and follow the installation steps:**

```
# Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Install Spark (modify version number if neccessary)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

# Unzip the Spark file in the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

# Set the Spark directory path to your system's environment variables.

import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

# Install findspark using pip
!pip install -q findspark
```

Step 2: Configure the Spark session.

We begin by configuring a Spark session using PySpark. This session will enable us to work with big data and leverage powerful analytical functionalities.

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("ExempluAplicatie2").getOrCreate()
```

Step 3: Upload de Data

Load wastewater quality data and associated characteristics into PySpark DataFrames.

```
# Load wastewater quality data into a DataFrame PySpark
wastewater_df = spark.read.csv("wastewater_data.csv", header=True,
inferSchema=True)

# Load associated characteristics of wastewater (e.g., temperature, pH,
conductivity)
features_df = spark.read.csv("water_features.csv", header=True,
inferSchema=True)
```

Step 4: Prepare the data

Perform data preparation, including feature processing and labeling for classification.

```
from pyspark.ml.feature import VectorAssembler

# Define the feature columns for vectorization
feature_columns = ["Temperature", "pH", "Conductivity"]

# Create an assembler to combine the features into a single column
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")

# Apply the assembler to the features
data = assembler.transform(features_df)

# Label the data: "Good" or "Low" for wastewater quality
from pyspark.sql.functions import when
wastewater_df = wastewater_df.withColumn("QualityCategory",
    when(wastewater_df["WastewaterQuality"] >= 80, "Good").otherwise("Low")
)
```

Step 5: Implement the classification model

Define a classification model, such as Random Forest, and train it to classify wastewater as "Calitate Bună" or "Calitate Scăzută" (“Good Quality” or “Low Quality”)

```
from pyspark.ml.classification import RandomForestClassifier

# Define the classification model
```

```
rf = RandomForestClassifier(featuresCol="features",
labelCol="QualityCategory", numTrees=10)

# Split the data into training and testing sets
train_data, test_data = data.randomSplit([0.7, 0.3])

# Train the model on the training data
model = rf.fit(train_data)
```

Step 6: Evaluate the model performance

Evaluate the performance of the classification model using metrics such as precision, recall, and F1-score.

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Make predictions on the test data
predictions = model.transform(test_data)

# Use an evaluator for classification metrics
evaluator = MulticlassClassificationEvaluator(labelCol="QualityCategory",
metricName="accuracy")

# Calculate the accuracy of the model
accuracy = evaluator.evaluate(predictions)
print(f"Precizia modelului: {accuracy}")
```

```
# Stop the Spark session
spark.stop()
```

Step 7: Interpret the results and explain the operations that have been performed.

Laboratory – Application 3

Objectives:

- Understanding big data analysis techniques applicable to experimental datasets collected from wastewater treatment plants.
- Mastering specific terms and understanding the importance of online monitoring as a crucial aspect of digitizing wastewater treatment plants.
- Data integration, machine learning, and predictive modeling.

Introduction:

In this laboratory, we will focus on using PySpark for the analysis of wastewater quality data and the development of a linear regression model capable of predicting wastewater consumption based on observed characteristics.

Step 1: Open Google Colab (https://colab.research.google.com/) **and follow the installation steps:**

```
# Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Install Spark (modify version number if neccessary)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-
hadoop3.2.tgz

# Unzip the Spark file in the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

# Set the Spark directory path to your system's environment variables.

import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

# Install findspark using pip
!pip install -q findspark
```

Step 2: Configure the Spark session.

We begin by configuring a Spark session using PySpark. This session will enable us to work with big data and leverage powerful analytical functionalities.

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("ExempluAplicatie2").getOrCreate()
```

Step 3: Upload de Data.

Load wastewater quality data and associated characteristics into PySpark DataFrames.

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("ExempluAplicatie2").getOrCreate()

# Load wastewater quality data into a DataFrame PySpark
wastewater_df = spark.read.csv("wastewater_data.csv", header=True,
inferSchema=True)

# Load associated characteristics of wastewater (e.g., temperature, pH,
conductivity)
features_df = spark.read.csv("water_features.csv", header=True,
inferSchema=True)
```

Step 4: Prepare the data.

Perform data preparation, including feature processing and labeling for classification.

```
from pyspark.ml.feature import VectorAssembler

# Define the feature columns for vectorization
feature_columns = ["Temperature", "pH", "Conductivity"]

# Create an assembler to combine the features into a single column
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")

# Apply the assembler to the features
data = assembler.transform(features_df)

# Define the label column (wastewater quantity)
label_column = "WastewaterFlow"
```

Step 5: Implement the Linear Regression model.

Define a linear regression model and train it to predict the amount of wastewater.

```
from pyspark.ml.regression import LinearRegression

# Define the linear regression model
lr = LinearRegression(featuresCol="features", labelCol=label_column)

# Split the data into training and testing sets
train_data, test_data = data.randomSplit([0.7, 0.3])
# Train the linear regression model
model = lr.fit(train_data)
```

Step 6: Make predictions and evaluate the performance.

Use the trained model to make predictions on the test data and assess the model's performance.

```
# Make predictions on the test data
predictions = model.transform(test_data)

# Use an evaluator for regression metrics
from pyspark.ml.evaluation import RegressionEvaluator

evaluator = RegressionEvaluator(labelCol=label_column,
predictionCol="prediction", metricName="rmse")

# Calculate the mean squared error
rmse = evaluator.evaluate(predictions)

print(f"Rădăcina Eroare Medie Pătratică (RMSE): {rmse}")
```

```
# Stop the Spark session
spark.stop()
```

Step 7: Interpret the results and explain the operations that have been performed.

Automatic control of processes
- course notes -

1. Introduction

What is automatic control of processes?

Systems engineering is the discipline that studies human-designed systems with the goal of manipulating their responses. Systems engineering is an interdisciplinary science that plays a crucial role in many engineering disciplines such as electrical engineering, mechanical engineering, chemical engineering, biotechnology, etc. There is a theoretical foundation that can be applied to all these systems despite substantial differences in their physical characteristics.

Automated process control is a branch of systems engineering that deals with controlling production facilities in industries such as chemical, petrochemical, food, biotechnology, etc. Automated process control plays a crucial role in the proper operation of facilities in terms of safety, product quality, and profitability. Even though biotechnological processes (e.g., biological wastewater treatment) have different physical characteristics than a robot, drone, or rocket, the underlying principles of systems theory are the same.

Systems theory is present in everyday life. Cars, refrigerators, washing machines, buildings, etc., have numerous automated control systems. What is impressive is that these automated control systems operate so efficiently that we barely notice their existence. They are reliable and make our lives better and safer, sometimes even more affordable.

The volume of knowledge generated in recent decades in the field of systems engineering is considerable, and the use of this knowledge for the design and operation of automated control systems is vital for the development of bio(processes).

Control systems in closed-loop. Terminology.

In this section, essential concepts and terminology used in the closed-loop control of a process will be presented (Kravaris & Kookos, 2021). We will take the example of the "primitive" control system of a liquid storage tank, as depicted in Figure 1.1. In this setup, a liquid flow feeds the storage tank (process), and an operator (regulator/controller) seeks to maintain the level in the tank (measured/controlled variable) at a desired value (setpoint), using a logical procedure (control algorithm) based on their experience. The mechanism through which the operator can maintain the level of the storage tank involves opening or closing a valve (*final control element*) that adjusts the feed flow rate (*manipulated variable*).

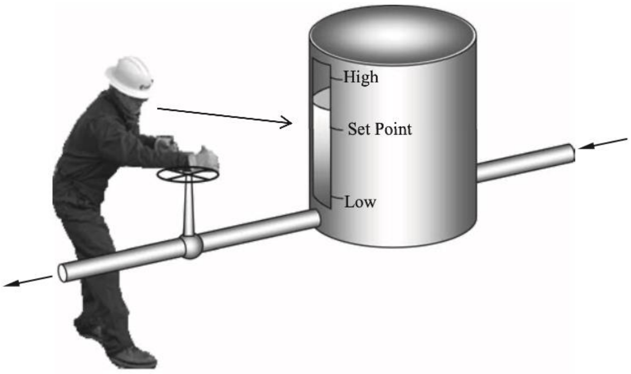
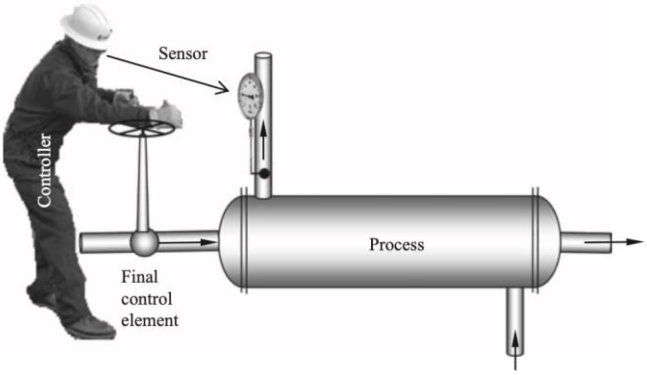


Figure 1.1. „Primitive” Level Control System for a Storage Tank

Oftentimes, processes are considered to operate in a steady-state. This is a logical simplification that allows engineers to design fairly complex processes in a reasonable amount of time. However, in reality, processes operate in a dynamic environment. Let's imagine that we are designing a heat exchanger that uses seawater as a cooling medium to reduce the temperature of a process stream from 100°C to 50°C. In the design phase, we need to make an estimate regarding the temperature of the cooling water (a single value), and let's assume we chose a temperature of 20°C. Now, think about the chances of the actual temperature of the cooling water being exactly 20°C. Will the system fail if the actual water temperature is 15°C or 10°C? The answer is yes; the system will fail to maintain the temperature at the desired value of 50°C unless a valve is installed that can appropriately adjust the flow rate of the cooling water. Additionally, a temperature sensor needs to be installed to measure the temperature of the outgoing stream from the heat exchanger. Then, using the measured and recorded temperature, an operator can check if the temperature is at the correct value and adjust the flow rate of the cooling water accordingly to correct any discrepancies, as illustrated in Figure 1.2. The seawater temperature can vary throughout the day, so the operator will need to make frequent adjustments to the valve to keep the stream temperature close to the desired temperature.



Figurae 1.2. „Primitive” system of temperature control

The main types of variables encountered in this "primitive" control system are as follows:

- **Disturbances**: Any process is affected by numerous external factors, many of which vary in an uncontrollable and unpredictable manner. These are disturbances that cause the process to deviate from the setpoint. In the illustrated heat exchanger case, disturbances could be the temperature of the cooling water, the temperature and flow rate of the input stream, or equipment aging that may increase heat transfer resistance. Some disturbances can be measured in real-time, but others are difficult, costly, or even impossible to measure.

- **Manipulated Variables**: Manipulated variables are process variables adjusted by the controller to achieve control objectives. A manipulated variable is also called an input variable or control variable to indicate that it represents the control action that "enters" the process. One of the most common input variables in a wastewater treatment plant is the feed flow rate, which can be manipulated using a valve or pump.

- **Measured Variables**: Measured variables are all variables for which sensors or measuring devices are installed to continuously measure and transmit the current value of the variable. Examples of measured variables in a wastewater treatment plant include temperature, flow rate, level, pH, dissolved oxygen, etc. A measured variable that a controller maintains at a desired value is called a controlled variable or output variable. The desired value at which the output variable is to be maintained is called the setpoint. This value is usually kept constant for a long period, but sometimes there may be a need for changes, and this must be done by a controller. If there is a difference between the setpoint and the controlled variable, we say there is an error. Mathematically, the error is defined as the difference between the setpoint and the controlled variable, and the controller's objective is to make the error zero.

In Figure 1.3, the general diagram of a closed-loop control system is presented, where all types of variables participating in the process, as well as the basic elements of the control system and how they are interconnected, can be observed.

The final control element (usually a valve or pump), along with the process and sensor, forms the physical system or open-loop system. We can see in Figure 1.3 that when the sensor is connected to the controller, and the controller acts on the final control element, the entire system has a circular structure, like a loop, and is called a closed-loop system.

A closed-loop control system involves continuous monitoring of the controlled variable and using it to make adjustments in the process through changes in the manipulated variable. The action of the controller is usually based on error, meaning the difference between the setpoint and the output variable. Depending on the error (its current value, history, and trend), the controller takes corrective actions. In simple terms, the operation of a closed-loop control system can be described as follows: monitoring, detection, and correction.

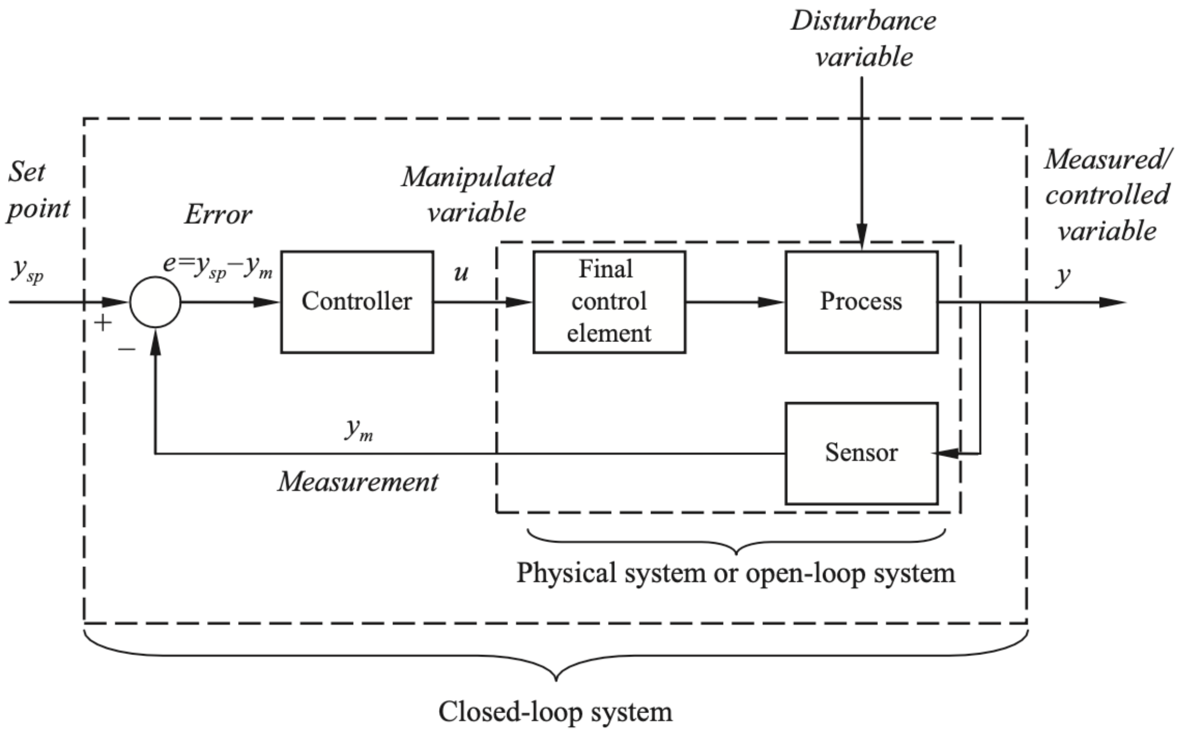


Figure 1.3. The elements of a closed-loop control system and and their interconnections (Kravaris & Kookos, 2021)

The standard notations used in process control are also indicated in Figure 3. The value of the measured and controlled variable is denoted by y. The measurement is noted as y_m, and it may not correspond to y in a transient state, as the sensor signal may have a delay compared to the modification of the physical variable it measures. The desired value or reference value of the controlled variable is denoted by y_{sp}. The error signal $e = y_{sp} - y$ is also indicated in the diagram (the small circle with two inward arrows with corresponding signs and an outward arrow indicates the subtraction operation). The error signal e acts on the controller, which decides the corresponding adjustments to correct the error and, ultimately, bring it back to zero. The signal u from the controller sets the value of the manipulated variable of the process, which is actually implemented by the final control element. Finally, the sensor detects the change in the system's response, and the loop is closed.

2. Mathematical Modeling of (Bio)Processes through Dynamic Mass Balance Equations

A mathematical model is a representation of our knowledge about a physical system, translated into a set of mathematical equations. The purpose is to use the model to improve our understanding of the real system's behavior (simulating the model can provide information) and to design and optimize process operation.

Dynamic models provide a quantitative description of the transient behavior of a process, in addition to its characteristics in a steady state. This is very useful for selecting appropriate operating conditions for a process, avoiding unwanted transient states. It is also crucial for controller design, as a controller can modify the dynamic behavior of a process, for better or for worse.

Like other engineering processes, biotechnological processes are subject to universal laws, such as the law of mass conservation, which states that in a closed system with no mass and energy transfer, its mass remains constant over time. Matter cannot be created or destroyed, although it can be rearranged in space or take on another form. This fundamental law underlies the development of mass balance equations (or material balance) and serves as an important tool in the study of (bio)processes.

The mass balance is a simple way to quantify the mass of a system at a certain moment in time:

$$\begin{pmatrix}\text{System mass}\\ \text{at time } t+\Delta t\end{pmatrix} = \begin{pmatrix}\text{System mass}\\ \text{at time } t\end{pmatrix} + \begin{pmatrix}\text{Mass entering the}\\ \text{system at time}\\ t+\Delta t\end{pmatrix} - \begin{pmatrix}\text{Mass exiting the}\\ \text{system at time}\\ t+\Delta t\end{pmatrix} \tag{2.1}$$

Equation 2.1 is useful when the beginning and end of the period for which the mass balance is calculated are well-known, as each term in the equation is expressed in mass units (e.g., g, kg, tons). However, in engineering sciences, it is preferred to express the mass being transported through the system as mass flow rate, given in mass units per unit of time (e.g., kg/h). For this purpose, Equation 2.1 is divided by Δt and rearranged:

$$\frac{\begin{pmatrix}\text{System mass}\\ \text{at time } t+\Delta t\end{pmatrix} - \begin{pmatrix}\text{System mass}\\ \text{at time } t\end{pmatrix}}{\Delta t} = \frac{\begin{pmatrix}\text{Mass entering the}\\ \text{system at time}\\ t+\Delta t\end{pmatrix}}{\Delta t} - \frac{\begin{pmatrix}\text{Mass exiting the}\\ \text{system at time}\\ t+\Delta t\end{pmatrix}}{\Delta t} \tag{2.2}$$

It can be observed that the term on the left-hand side represents the rate of accumulation of mass in the system (the rate at which mass is accumulating in the system), and the terms on the right-hand side represent the mass flow rate (the mass transported through the system per unit of time). In other words:

$$\begin{pmatrix}\text{Rate of mass}\\ \text{accumulation in the system}\end{pmatrix} = \begin{pmatrix}\text{Input mass}\\ \text{flow rate}\end{pmatrix} - \begin{pmatrix}\text{Output mass}\\ \text{flow rate}\end{pmatrix} \tag{2.3}$$

In this equation, the term associated with the rate of mass accumulation in the system represents the rate of change of the total mass in the system with respect to time. If the mass flow rate out is greater than the mass flow rate in, the mass at $t + \Delta t$ will be less than the mass at time t, and the accumulation rate will be negative, indicating "negative accumulation." Even if the rate of mass accumulation in the system is negative, the total mass of the system cannot be less than 0.

When there is no mass accumulation in the system (the accumulation rate is equal to 0), it is said that the system is in a steady state, and Equation 2.3 can be written as follows:

$$\begin{pmatrix}\text{Input mass}\\ \text{flow rate}\end{pmatrix} = \begin{pmatrix}\text{Output mass}\\ \text{flow rate}\end{pmatrix} \tag{2.4}$$

Quantifying the total mass in a biotechnological system is useful only if it is done for elements that do not undergo transformation during the process. For example, a mass balance can be written for atomic elements:

$$\begin{pmatrix}\text{Rate of mass}\\ \text{accumulation of carbon}\\ \text{in the system}\end{pmatrix} = \begin{pmatrix}\text{Input mass flow}\\ \text{rate of carbon}\\ \text{in the system}\end{pmatrix} - \begin{pmatrix}\text{Output mass flow}\\ \text{rate of carbon}\\ \text{in the system}\end{pmatrix} \tag{2.5}$$

The majority of components of interest in a biotechnological process undergo transformations: biomass increases as it feeds on substrates, products form under the influence of biomass, etc. A component may not be transported from outside the system but may form inside the system from other components that, in turn, have been brought in from outside the system. In conclusion, when quantifying the mass of any component undergoing transformations, it is necessary to take into account the production/consumption rate through reactions. The mass balance written for a component i of the system can be expressed as follows:

$$\boxed{\begin{gathered}\begin{pmatrix}\text{Rate of mass}\\ \text{accumulation of a}\\ \text{component in the sistem}\end{pmatrix} = \begin{pmatrix}\text{Input mass flow}\\ \text{rate of a component}\\ \text{in the system}\end{pmatrix} - \\ \begin{pmatrix}\text{Output mass flow}\\ \text{rate of a component}\\ \text{in the system}\end{pmatrix} + \begin{pmatrix}\textbf{Rate of production}\\ \textbf{of a component}\\ \textbf{through reaction}\end{pmatrix}\end{gathered}} \tag{2.6}$$

For simplification, we used "+" for "production," understanding that for the consumption of component i (e.g., consumption of food/substrate for the growth of microorganisms), the term will be negative. We will use Equation 2.6, in this form, as the general mass balance relationship for components, where the production term through reaction can take negative values - "negative production." Equation 2.6 will be used for quantifying any component i in a biotechnological process.

The correct formulation of dynamic mass balance equations is crucial in the mathematical modeling of biotechnological processes. Therefore, the modeling methodology must be coherent and provide the necessary steps for their development.

Developing dynamic mass balance equations

A dynamic mass balance equation is a mathematical expression that describes the dynamics of numerous chemical or biological components being transported through a system. Literature provides several procedures for developing dynamic mass balance equations (Doran, 1995; Snape et al., 1995; Dochain, 2008), but regardless of the approach, understanding the process to be modeled is essential. The following are three steps that can be followed to develop mass balance equations for any component of a biotechnological process.

A. Choosing the balance area

The choice of a balance area depends on the objective of mathematical modeling and can encompass very small volumes (e.g., an atom, a molecule, a cell, etc.) or extremely large volumes (e.g., a continent, a planet, a solar system, etc.). The usual balance areas for biotechnological processes are typically limited to a bioreactor or a facility containing multiple tanks but can be narrowed down to a gas bubble, an activated sludge floc, or a microorganism.

When establishing the boundaries of the balance area, consider the complexity of the system within the balance area. If a modeled component undergoes transformations at multiple points in the system, it can be divided into subsystems that will be modeled individually. Thus, the mass of the component within the chosen balance area must be constant or quasi-constant (so that the difference between the mathematical model and the measured quantities is acceptable).

Figures 2.1 and 2.2 illustrate the diversity of balance areas. The control volume can be continuous (Figure 2.1 – an example of a reactor) or discontinuous (Figure 2.2 – the leaves of a tree). Modeling is done similarly for both types of control volumes.

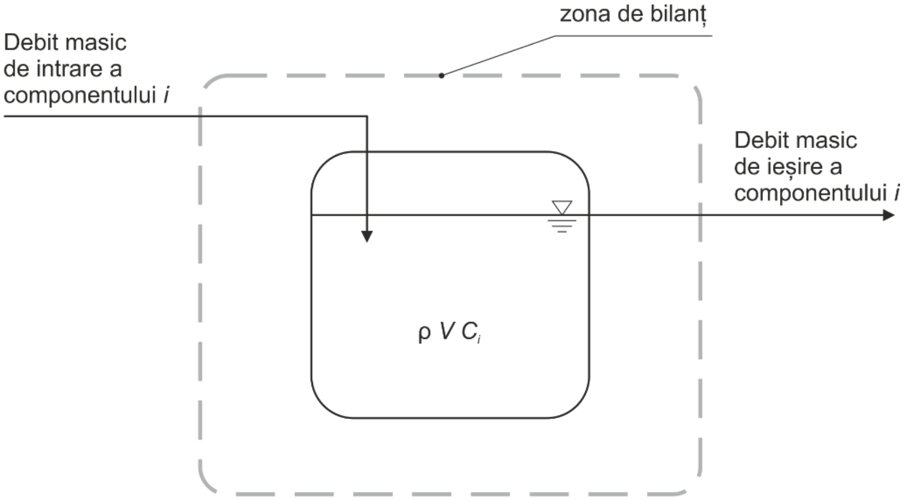

zona de bilanț
Debit masic de intrare a componentului *i*
Debit masic de ieșire a componentului *i*
$\rho V C_i$

Figura 2.1. Zona de bilanț a unui reactor

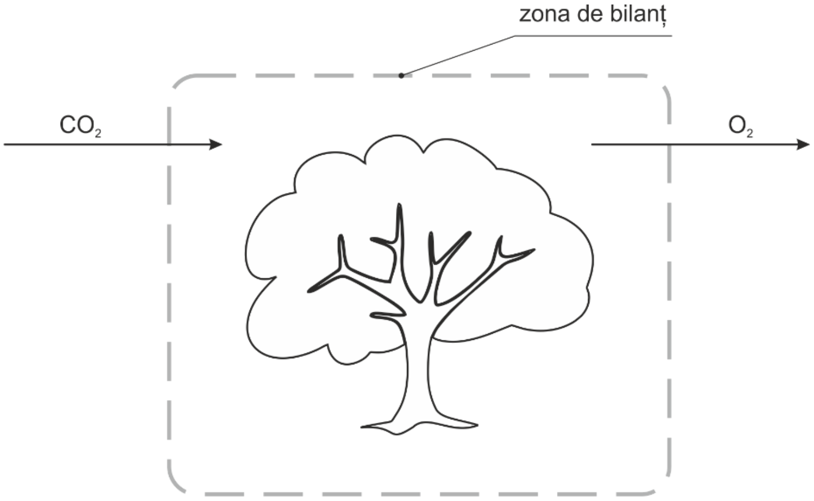


Figura 2.2. Zona de bilanț a unui arbore

After choosing the balance area, the components to be modeled must be established. In the case of a biotechnological process, the chemical or biological species subject to investigation will be determined. There are numerous compounds that can be the subject of the mass balance, such as:

- Microbial biomass (e.g., a specific species, a class of microorganisms, activated sludge from wastewater treatment processes),
- Substrates (e.g., the carbon source for growth, organic pollutants in wastewater expressed by chemical oxygen demand - COD, macro-nutrients such as nitrogen and phosphorus, etc.),
- Reaction products (e.g., enzymes, organic acids, proteins, carbohydrates, etc.),
- Dissolved gases (e.g., dissolved oxygen, dissolved carbon dioxide, etc.),
- Populations of organisms (e.g., microorganisms, plants, animals, humans, etc.),
- Other compounds of interest (e.g., pesticides, antibiotics, etc.).
- Biotechnological processes are, in fact, processes that occur in nature (biological processes) but are intensified in special installations called bioreactors. Thus, the usual choice for the balance area when modeling biotechnological processes is the bioreactor. Often, biotechnological installations consist of multiple bioreactors associated with basins that serve other operations (e.g., chemical preparation, storage, settling, degassing, etc.). These complex installations are divided into subsystems that are modeled individually.

Modeling biotechnological processes needs to be simplified to describe the dynamics of process state variables as easily as possible. State variables are the components that characterize the process and can be measured or estimated. Regarding the distribution of a component in a bioreactor, they can be:

- Homogeneous – reactors with complete mixing. They are systems with concentrated parameters, described by ordinary differential equations where the derivative variable is time, t.
- Heterogeneous (with a concentration gradient) – tubular or piston-type reactors. They are systems with distributed parameters, described by partial differential equations

where another derivative variable, in addition to time, is involved (a spatial coordinate: depth, length).

The reactor with complete mixing. In a reactor with complete mixing, the distribution of the component to be modeled is uniform at any point within its volume. Continuous mixing reactors can be of three types:

- Batch reactors – all the substrate is added from the beginning, it is inoculated, and throughout the process, no fresh substrate is added, and no reactor content is removed. In other words, there are no mass inflows or outflows. The accumulation rate of component i in the bioreactor (equation 2.6) results only from the last term, the rate of production of the component through the reaction.
- Semi-batch reactors – at the beginning of the process, only a small volume of substrate is inoculated, and throughout the process, fresh substrate is added without removing the bioreactor contents. Feeding to the bioreactor stops when the maximum useful volume is reached. There is a mass inflow, but there is no mass outflow.
- Continuous reactors – the bioreactor is continuously fed and operates at the maximum useful volume. This means that the mass outflow rate is equal to the mass inflow rate.

Figure 2.3 illustrates a continuous bioreactor with complete mixing. The concentration of any component in the mass outflow is equal to its concentration inside the bioreactor. The total mass in the system M (kg) is given by the product of the tank volume V (m^3) and the density of the liquid in the tank ρ ($kg \cdot m^{-3}$), $M = V\rho$. The mass of any component i in the bioreactor is expressed in mass units or as the number of moles, by taking the product of the volume V and the concentration of component i, C_i (kg/m^{-3} or $kmoli/m^{-3}$), so $i = C_i V$ will be expressed in kg or kmol.

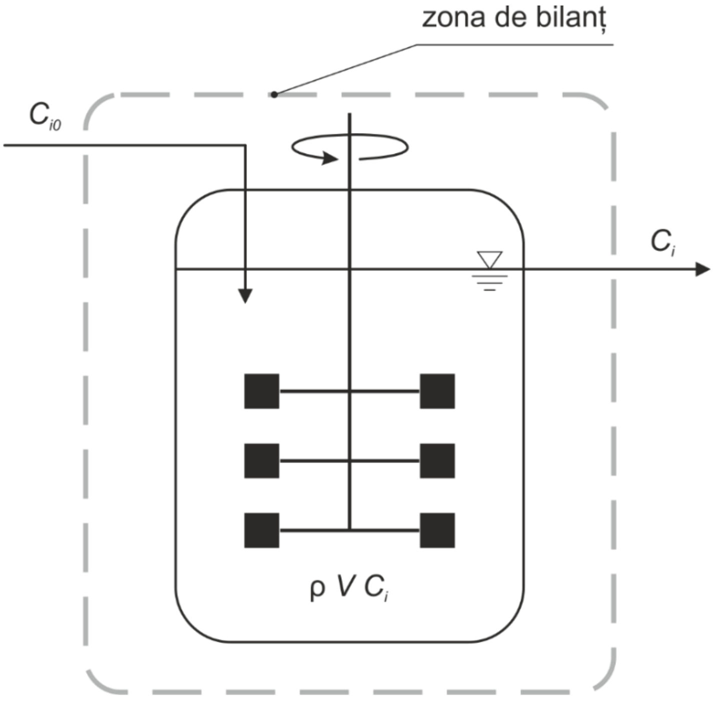


Figure 2.3. The balance area around a reactor with complete mixing

The reactor with a concentration gradient. A reactor with a concentration gradient is characterized by the fact that the distribution of the component to be modeled is not uniform at any point within its volume but varies along one of its dimensions (height or length). An example of this could be a settler that has a nearly zero particle concentration at the top and a very high concentration at the bottom.

This type of reactor is an important concept in bioprocess modeling. In a tubular reactor, the concentrations of the modeled components vary along the reactor (or in height if a settler is modeled) even when it is operated under steady-state conditions. In an ideal tubular reactor, the concentration of the components is constant transversely in any zone of the reactor but varies axially. In other words, the concentration of a component i is constant along one of the dimensions (ox or oy) but varies along the other dimension (concentration gradient).

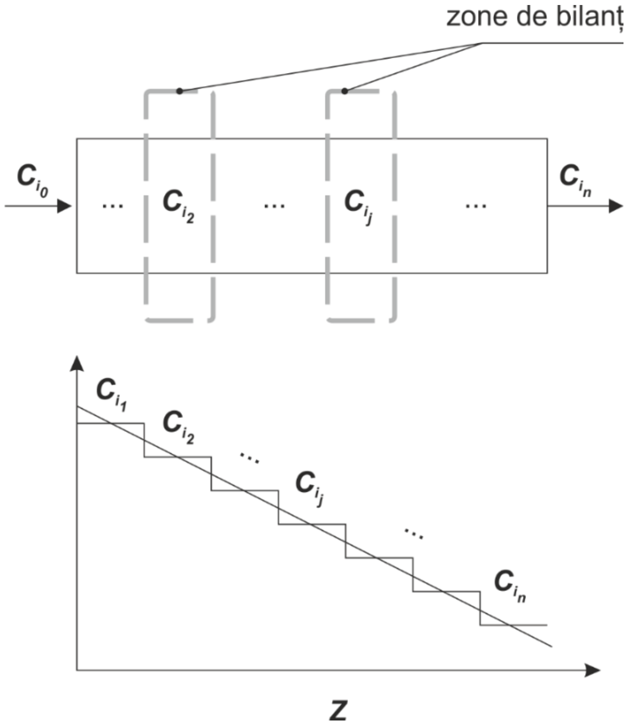


Figure 2.4. The balance area for a piston-type reactor and the approximate concentration gradient

Figure 2.4 presents the diagram of a tubular reactor where the concentration of a component i decreases from the inlet to the outlet. This reactor is modeled by partial differential equations (*t* and *z* are the derivative variables). However, the concentration of component i can be considered constant if a small enough balance area is chosen, so that its concentration is quasi-constant. As long as the concentration of component i can be considered constant, modeling will be done through ordinary differential equations, with the only derivative variable being time. Thus, a tubular reactor can be divided into n balance areas, whose concentration $C_{i_j, j=\overline{1,n}}$will be modeled by n ordinary differential equations. The n balance areas can be equal, and then we talk about systems with uniformly distributed parameters or unequal, and the systems will have non-uniformly distributed parameters. The choice of the size of the balance area depends heavily on the nature of the modeled component. By dividing the tubular reactor into several areas where the concentration of the components is uniform, it can be stated that the volume of a tubular reactor is formed from several completely mixed reactors.

Often, a biotechnological installation is made up of combinations of completely mixed and concentration gradient reactors, but there are also cases where, in the same reactor, the concentration of one component is uniform throughout the reactor volume, and the concentration of another component has a gradient, for example:

- In the biological treatment of wastewater, an aerated basin with complete mixing is connected in series with a settler for separating activated sludge from treated water.
- In a photobioreactor with complete mixing, all components have uniform concentration, but light is attenuated within the culture.

B. Identifying transport flows

After choosing the balance area, the second important step in developing dynamic mass balance equations is identifying the transport flows across the system boundaries.

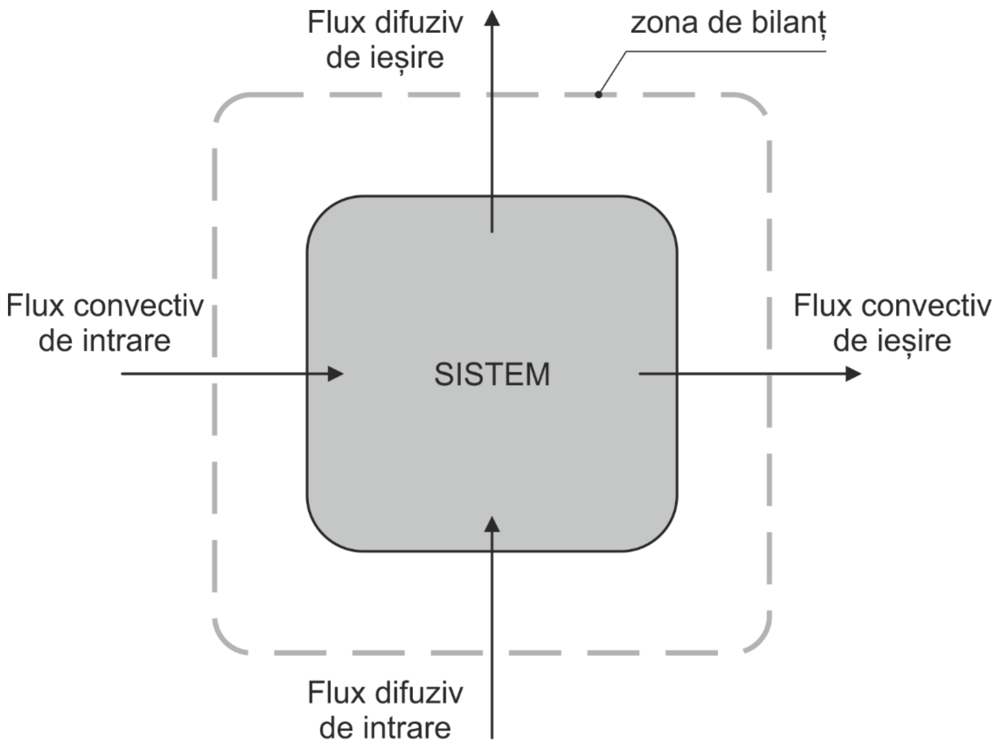


Figure 2.5. The balance area and the input and output flows

These flows can be well-defined physical rates and can be divided into (Figure 2.5):

- Convective flows – are liquid flows that transport various dissolved or colloidal components. These can be identified at the system's inlet (e.g., the flow of fresh substrate) and at the outlet (e.g., the flow of treated water leaving the aerobic basin).
- Diffusive flows – are gas flows that are bubbled into the culture of microorganisms. An incoming diffusive flow has two components: the actual gas flow and the interfacial transfer rate. Mixtures of gases (e.g., air) with different transfer rates can be bubbled (nitrogen is inert while oxygen dissolves easily). Outgoing diffusive flows are often ignored because they are not of interest. For example, it is important to know the air flow and the mass transfer rate of oxygen bubbled into an aerobic process, but information about the exhausted gas leaving the basin is unnecessary.

The terms associated with input flows are positive, and the terms associated with output flows are negative.

C. Mathematical computing

In this final stage, each term of the generalized mass balance equation (equation 2.6) will be expressed mathematically. The resulting equations should contain measurable or estimable quantities. For simplicity, we will use the following equation:

$$(\text{Accumulation}) = (\text{Input}) - (\text{Output}) + (\text{Production}) \tag{2.7}$$

The advantage of a balance equation is that it has a logical basis, and the mathematical expressions will maintain this basis.

The term associated with the rate of accumulation. The mass of an open system or a component of the system can change over time, and the rate of accumulation is mathematically expressed as the derivative of mass with respect to time:

$$\begin{pmatrix} \text{Mass accumulation rate} \\ \text{of a component } i \text{ in the system} \end{pmatrix} = \left(\frac{dM_i}{dt}\right) \tag{2.8}$$

where M_i is the mass of component i in the system and can be expressed in kg or molar units, and time can be expressed in any unit (seconds, hours, etc.).

As mentioned earlier, in engineering sciences, the use of concentration for a component is preferred over mass. To convert mass to concentration, knowledge of a reference volume is necessary:

$$\frac{dM_i}{dt} = \frac{d(V\,C_i)}{dt} \tag{2.9}$$

where C_i is the concentration of component *i* expressed in mass units per volume unit (e.g. kg/L, kmol/L etc.).

For a component *i* in the system, which is present in gaseous form, the ideal gas law is used, expressing the concentration in terms of partial pressure and molar fraction:

$$p_i V = n_i RT \tag{2.10}$$

where p_i is the partial pressure of component i in the gas phase, *R* is the ideal gas constant, *n* is the number of moles, and *T* is the temperature. The molar concentration of the gaseous component can be expressed as::

$$C_i^m = \frac{n_i}{V} = \frac{p_i}{RT} = \frac{y_i P}{RT} \tag{2.11}$$

where y_i is the molar fraction of component *i* in the gas phase, and *P* is the total pressure of the system. The molar fraction is dimensionless, and the sum of the molar fractions of a gas mixture is equal to 1. Thus, the term associated with the accumulation rate for a gaseous component can be expressed in molar concentration or molar fraction:

$$\frac{dn_i}{dt} = \frac{d(V\ C_i^m)}{dt} = \frac{d\left(\frac{p_i V}{RT}\right)}{dt} = \frac{d\left(\frac{y_i PV}{RT}\right)}{dt} \tag{2.12}$$

The terms associated with convective transport fluxes. Mass flow rates are the product of volumetric flow rates and density:

$$(\text{Debit masic convectiv}) = (\text{Debit volumetric})\left(\frac{\text{Masă}}{\text{Volum}}\right) \tag{2.13}$$

If we compute the mass of a component as $M_i = VC_i$, by replacing it in equation 2.13, then the mass flow rate of a component *i* is:

$$M_i = FC_i \tag{2.14}$$

where *F* is the volumetric flow rate measured in units of mass per volume (e.g. mL/min, L/h etc.).

Modeling diffusive fluxes is somewhat more complex. Mass balance equations written for gas components dissolved in liquid have an additional term that describes gas-liquid mass transfer. Sometimes, these terms explicitly include the molar inflow rate. For oxygen, this term can take the following form:

$$N_{O_2} = k_L a\left(C_{O_2}^* - C_{O_2}\right) \tag{2.15}$$

where N_{O_2} is the gas-liquid mass transfer rate, $k_L a$ is the gas-liquid mass transfer coefficient, $C^*_{O_2}$ is the oxygen saturation concentration, and C_{O_2} is the current oxygen concentration. $k_L a$ can be modeled to explicitly include the molar flow rate of oxygen (diffusive inflow flux).

Diffusive outflow fluxes are rarely measured but result from the difference between the diffusive inflow flux and the gas-liquid mass transfer rate.

The term associated with the production rate allows the expression of the production or consumption of a component through a chemical or biological reaction (e.g., the growth rate of microorganisms, substrate consumption rate, the rate of formation of a reaction product). In biotechnological processes, the production rate of a component is expressed in units of mass per unit volume per time (e.g. g/L/h):

$$\begin{pmatrix}\text{The mass production}\\ \text{rate of component } A\end{pmatrix} = \begin{pmatrix}\text{The volumetric}\\ \text{production rate}\end{pmatrix}\begin{pmatrix}\text{System}\\ \text{volume}\end{pmatrix} \qquad (2.16)$$

$$R_i = r_i V \qquad (2.17)$$

where R_i is the mass reaction rate, and r_i is the volumetric reaction rate. Equivalent molar quantities can also be used (e.g., the molar rate of total inorganic carbon). The volumetric reaction rate is positive for production and negative for consumption.

The general dynamic model.

The mass accumulation rate of any component *i* within a system (equation 2.6), expressed considering the mass balance, regroups two types of terms:

- conversion terms (describing the kinetics of chemical and biological reactions and conversion yields)
- transport terms (describing the mass transition through processes, in liquid and/or gaseous state, and phase transfer phenomena) (Sablani et al., 2006; Dochain, 2008).

The change over time in the mass accumulation rate of a component i in a system is most often expressed using the concept of concentration (equation 2.9).

A series of hypotheses can be formulated for a biotechnological process:

- the process takes place in a completely stirred reactor (note: a bioreactor with a concentration gradient is divided into n completely mixed reactors), only convective flows exist, not diffusive ones (no gas bubbling in the reactor),
- there are n convective inflow streams.

Under these conditions, a general dynamic mass balance equation can be defined as follows:

$$\frac{dVC_i}{dt} = r_i V + \sum_{j=1}^{n} F_{in,j} C_{i,0} - F_{out} C_i \qquad (2.18)$$

where $C_{i,0}$ is the concentration of component i in the feed stream, and r_i is the volumetric production or consumption rate of component i. The conversion term, r_iV, describes the production or consumption of a component through chemical or biological reactions (e.g., biomass growth rates, substrate assimilation, product formation, etc.), while the other two terms describe the convective transport flows of component i across the system boundaries. The general dynamic model (2.18) can be simplified since the presence of a component in more than one input stream, F_{in}, is very rare, allowing the replacement of the input term with $F_{in}C_{i,0}$. The output flow, F_{out}, of a completely mixed reactor has properties identical to those of the fluid inside the reactor.

3. Case study. Modeling the aerobic biological treatment process of wastewater.

Biological treatment of wastewater is more than a necessity; it is a responsibility of every producer who must continuously improve the process. There are several biological wastewater treatment processes, leading to numerous mathematical models. Wastewater biological treatment processes are highly complex, strongly nonlinear, and characterized by parametric uncertainties. We will further develop a simplified model for the aerobic biological treatment of wastewater to help us understand and simulate this process on a computer.

Description of the aerobic wastewater treatment process

The objective of the aerobic biological treatment process of wastewater is to transform the organic pollutants present in the wastewater into stable oxidized products and new biomass (activated sludge). Aerobic wastewater treatment is a process in which the organic pollutants in the wastewater are oxidized toCO_2, H_2O, NH_4^+ and new cells.

A wastewater biological treatment plant represents an assembly of distinct treatment processes or units that generate an effluent of a certain quality from an influent with known flow rate and composition.

Aerobic biological treatment of wastewater, also known as secondary treatment, utilizes microorganisms abundantly present in the natural environment to convert organic pollutants into dense microbial mass (activated sludge), which can be easily separated from the purified water through conventional sedimentation processes. The aerobic wastewater treatment process is mainly carried out by heterotrophic microorganisms capable of breaking down various organic pollutants through two different processes: oxidation and biosynthesis, both leading to their removal from the influent (wastewater).

Oxidation or respiration (equation 3.1) results in inorganic end products, while biosynthesis (equation 3.2) transforms soluble and colloidal organic substances into biomass, which can be removed through sedimentation. When the concentration of organic pollutants, serving as the food source for the microorganisms forming the activated sludge, becomes insufficient, cells enter a process of endogenous respiration to obtain the necessary energy for maintenance (auto-oxidation, equation 3.3). All three processes occur simultaneously in the aerobic basin and can be stoichiometrically expressed as follows (Gray, 2005):

Oxidation

$$\mathrm{COHNS} + \mathrm{O_2} + \mathrm{microorganisms} \rightarrow \mathrm{CO_2} + \mathrm{NH_3} + \mathrm{other\ final\ products} + \mathrm{energy} \qquad (3.1)$$

Biosynthesis

$$\mathrm{COHNS} + \mathrm{O_2} + \mathrm{microorganisms} \rightarrow \mathrm{C_5H_7NO_2} \qquad (3.2)$$

Auto-oxidation

$$\mathrm{C_5H_7NO_2} + 5\mathrm{O_2} \rightarrow 5\mathrm{CO_2} + \mathrm{NH_3} + 2\mathrm{H_2O} + \mathrm{energy} \qquad (3.3)$$

where COHNS and $\mathrm{C_5H_7NO_2}$ are the general equations for organic matter and for microorganisms.

To ensure the mineralization of organic pollutants, the liquid mixture in the aerobic basin is enriched with dissolved oxygen provided through air diffusion with the help of turbo blowers. Microorganisms that feed on these organic pollutants agglomerate, forming activated sludge flocs that have the ability to settle.

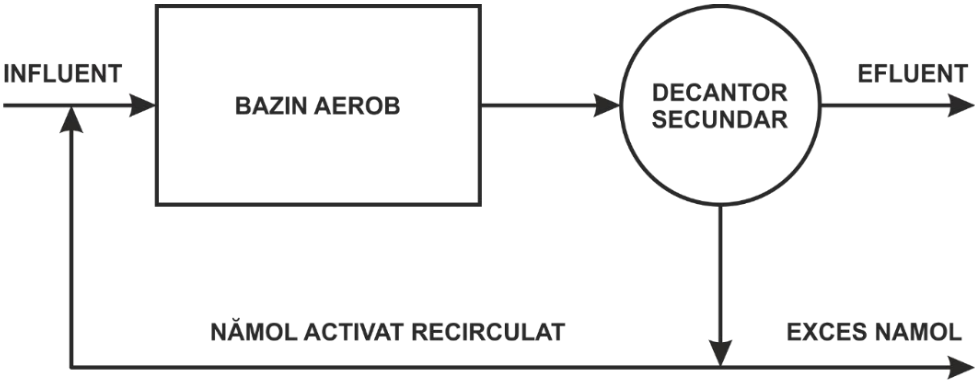


Figure 3.1. The conventional aerobic wastewater treatment system

Thus, a classic aerobic wastewater treatment process consists of three fundamental elements (Figure 3.1):

- **Aerobic basin:** This is where the oxidation of organic pollutants takes place. The influent of the aerobic basin (wastewater) is mixed with recirculated activated sludge that has settled in the clarifier, resulting in a longer retention time for the activated sludge in the facility compared to the hydraulic retention time. This allows maintaining a large number of microorganisms in the aerobic basin capable of efficiently oxidizing the pollutant organic matter in a short period.

- **Clarifier (Decantor):** Used for the sedimentation of flocs of activated sludge produced in the aerobic basin. Some of the settled activated sludge is recirculated, and the excess is removed.

- **Recirculation system:** Vital for wastewater treatment plants, it not only plays the role described above but also prevents the "wash-out" phenomenon that can occur when the influent flow is very high (e.g., during a rainy period).

Mathematical modeling of the aerobic basin:

As mentioned earlier, the growth of heterotrophic microorganisms populating the aerobic basin requires dissolved oxygen. However, their growth is influenced by the soluble substrate (organic pollutants in wastewater), which serves as their source of carbon and energy. Therefore, they are limited not only by oxygen but also by the availability of organic carbon, macro-nutrients (e.g., nitrogen, phosphorus, etc.), and other growth factors. Biomass growth is considered the main process, while other processes are coupled through stoichiometric parameters, meaning the other processes are proportional to the growth of activated sludge. The simplest mathematical model describing the aerobic basin is presented in Table 3.1. In accordance with the IWA (International Water Association) nomenclature, X denotes insoluble components, and S denotes soluble ones. The subscripts refer to individual components: biomass (B), substrate (S), and dissolved oxygen (O).

S_S represents the carbon source (organic substrate with carbon), often expressed as COD (Chemical Oxygen Demand), measured in mg $O_2 \cdot L^{-1}$.

Macro-nutrients (e.g., nitrogen, phosphorus, etc.) are not modeled here but are considered to be present in sufficient concentrations not to limit the growth of heterotrophic microorganisms.

Table 3.1. Kinetic Terms and Stoichiometric Parameters of the Aerobic Biological Wastewater Treatment Process

Component →	i	1	2	3	Process rate, ρ_j
j Process ↓		X_B	S_S	S_O	[g·L^{-1}·h^{-1}]
1 Growth		1	$-\frac{1}{Y}$	$-\frac{1-Y}{Y}$	$\mu_{max}\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{S_O}{K_O+S_S}\right)X_B$

2 Decline	-1			$\mu_d X_{\mathrm{B}}$
The observed conversion rates [g·L^{-1}·h^{-1}]	$r_i = \sum_j r_{ij} = \sum_j \nu_{ij}\rho_j$			*Kinetic parameters*: Maximum specific growth rate:: μ_{max} Half-saturation constant: K_S Specific decline rate: μ_d Dissolved oxygen saturation constant: K_{O}
Stoichiometric parameters, ν_{ij}: Growth yield: Y	Biomass [g·L^{-1}]	Substrate [g·L^{-1}]	Dissolved oxygen [g·L^{-1}]	

They can be modeled by simple stoichiometric equations, but are often overlooked because the typical C:N:P ratio of municipal wastewater is around 100:5:1, ensuring good growth of activated sludge and therefore effective purification capacity. If macro-nutrients exceed the recommended ratio, they cannot be removed through aerobic biological treatment alone. In such cases, additional basins are added to the biological treatment plant, capable of sustaining processes such as biological nitrogen removal through nitrification and denitrification or biological phosphorus removal.

Biomass growth is expressed through a dual limitation model of substrate and oxygen, and the decline rate is first-order with respect to the biomass concentration. The volumetric reaction rates resulting from Table 3.1 can be expressed as follows:

$$r_{X_{\mathrm{B}}} = \mu_{max}\left(\frac{S_{\mathrm{S}}}{K_{\mathrm{S}} + S_{\mathrm{S}}}\right)\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O}} + S_{\mathrm{S}}}\right)X_{\mathrm{B}} - \mu_d X_{\mathrm{B}} \tag{3.4}$$

$$r_{S_{\mathrm{S}}} = -\frac{1}{Y}\left(\mu_{max}\left(\frac{S_{\mathrm{S}}}{K_{\mathrm{S}} + S_{\mathrm{S}}}\right)\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O}} + S_{\mathrm{S}}}\right)X_{\mathrm{B}}\right) \tag{3.5}$$

$$r_{S_{\mathrm{O}}} = -\frac{1-Y}{Y}\left(\mu_{max}\left(\frac{S_{\mathrm{S}}}{K_{\mathrm{S}} + S_{\mathrm{S}}}\right)\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O}} + S_{\mathrm{S}}}\right)X_{\mathrm{B}}\right) \tag{3.6}$$

The growth of heterotrophic microorganisms occurs solely through the consumption of organic substances with carbon, whose complete mineralization only occurs in the presence of dissolved oxygen. Therefore, the second limitation factor of the form (i.e. $S_{\mathrm{O}}/(K_{\mathrm{O}} + S_{\mathrm{S}})$) is chosen based on mathematical criteria and no longer has the same meaning as the original Monod model. This factor serves as a continuous mathematical function with values between 0 and 1. The dissolved oxygen saturation constant K_{O}, usually has a small value, indicating that the oxygen limitation model is nearly 1 at moderate dissolved oxygen values but tends toward 0 as the dissolved oxygen concentration decreases to 0.

The volumetric rate of substrate removal and the volumetric rate of dissolved oxygen consumption are proportional to the volumetric growth rate of biomass (Equations 3.5 and 3.6).

As the concentration of dissolved oxygen is maintained in the aerobic basin through air diffusion, the gas-liquid mass transfer must be taken into account:

$$N_{\mathrm{O}_2} = (K_L a)_{\mathrm{O}_2}\left(S_{\mathrm{O,sat}} - S_{\mathrm{O}}\right) \tag{3.7}$$

where $(K_L a)_{\mathrm{O}_2}$ is the overall gas-liquid mass transfer coefficient, and $S_{\mathrm{O,sat}}$ is the concentration at dissolved oxygen saturation. The gas-liquid oxygen transfer rate can be expressed in terms of the aeration rate as follows:

$$(K_L a)_{\mathrm{O}_2} = \alpha W \tag{3.8}$$

where α is the diffusion coefficient referring to the oxygen transfer rate and thus the efficiency of the air diffusers, and W is the aeration rate.

Mathematical modeling of the clarifier (Ifrim, 2012):

The activated sludge flocs formed in the aerobic basin are allowed to settle in a separate basin, thereby clarifying the treated water. The liquid mixture entering the clarifier is divided into two streams (Figure 3.3), one stream leading to the upper part of the clarifier (overflow stream) and another stream extracted from the lower part (underflow stream). The gravitational settling of the activated sludge flocs occurs over these two streams, leading to the development of a concentration gradient throughout the depth of the clarifier. The concentration of activated sludge will be very low in the overflow stream and highly concentrated in the underflow stream. This depth and time-dependent concentration gradient create a system with distributed parameters, the mathematical modeling of which is quite complex.

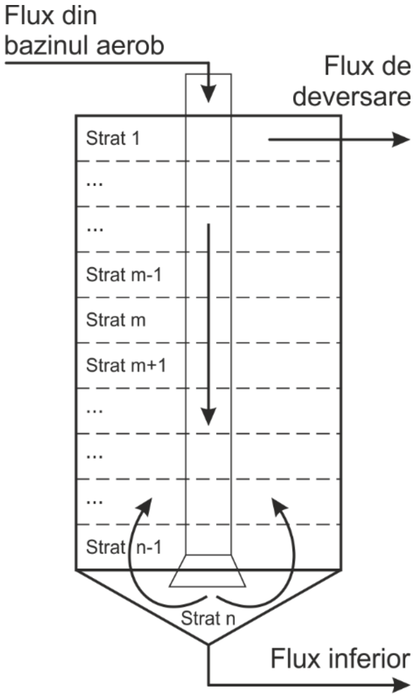


Figure 3.3. Multi-layer model of a vertical clarifier

The difficulty in modeling arises from the fact that the settling velocity of activated sludge flocs depends on numerous variables, such as sludge density, suspended solids concentration, aeration basin flow rate, size of sludge flocs, retention time, etc. A high concentration of activated sludge can lead to its compaction, a phenomenon that occurs at the base of the clarifier. In other cases, a low concentration of sludge flocs may result in inefficient settling. These phenomena are challenging to predict, and therefore, an empirical model can be adopted to generate a satisfactory response.

Figure 3.3 illustrates a multi-layer model of a vertical clarifier. The clarifier is divided into n layers, each being quasi-homogeneous. In other words, the clarifier (a piston-type basin) is divided into n completely mixed basins to avoid partial derivative models (in terms of depth and time). The objective of this model is to provide a satisfactory prediction of the concentration of settled activated sludge (which is different from the concentration of activated sludge in the aerobic basin) that is recirculated between the clarifier and the aerobic basin. The recirculated sludge influences all aerobic treatment process rates: the growth rate of activated sludge, substrate consumption rate, and dissolved oxygen consumption rate. For this purpose, only layer n will be modeled, as the others are not of interest (Figure 3.4).

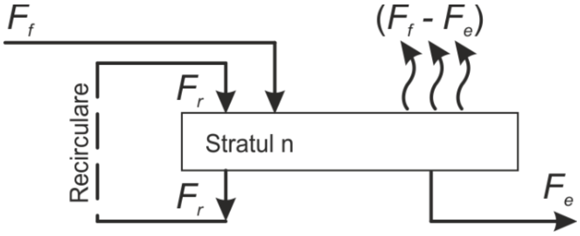


Figure 3.4. Mass balance at the base of the clarifier (layer n)

Since it is assumed that the kinetic rates are negligible in the clarifier (growth rate of activated sludge, substrate consumption rate, and dissolved oxygen consumption rate), only the biomass flows crossing the boundaries of the balance zone need to be identified (Figure 3.4). An equation can be written, taking into account the biomass transported through the influent flow from the aerobic basin (as a continuous process, this is equal to the influent flow rate), F_f, the recirculation flow, F_r, and the excess sludge flow F_e. The mass balance equation for the recirculated activated sludge in layer n takes the following form:

$$\frac{dX_{\mathrm{R}}}{dt} = \frac{F_f}{V_s}X_{\mathrm{B}} + \frac{F_r}{V_s}X_{\mathrm{B}} - \frac{F_e}{V_s}X_{\mathrm{R}} - \frac{F_r}{V_s}X_{\mathrm{R}} - \eta\frac{\left(F_f - F_e\right)}{V_s}X_{\mathrm{R}} \tag{3.9}$$

where X_{R} is the concentration of recirculated activated sludge, V_s is the volume of layer n, and η is a subunitary parameter quantifying the fraction of biomass migrating to layer n-1 with the overflow stream.

Application 1. Modeling and Control of the Level of a Storage Tank

Theoretical aspects

In the figure below, a cylindrical liquid storage tank is represented. The cross-sectional area is denoted as A, and its level is h. The useful volume of the tank is therefore $V = Ah$. The volumetric input flow rate is F_{in}, leading to the conclusion that the mass flow rate of input is $M_{in} = \rho F_{in}$. The volumetric output flow rate is F_{out} and thus the mass flow rate of output is $M_{out} = \rho F_{out}$, where ρ is the density of the liquid.

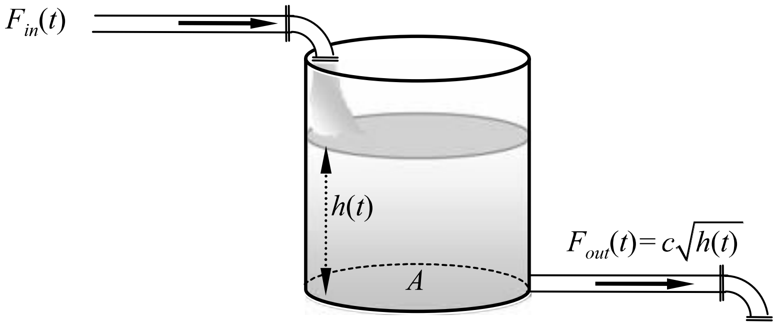


Figure 1. Liquid Storage Tank (Kravaris & Kookos, 2021)

The mass of the liquid in the storage tank is, therefore:

$$\frac{dM(t)}{dt} = M_{in}(t) - M_{out}(t) \tag{1}$$

or

$$\frac{d(\rho V(t))}{dt} = \rho F_{in}(t) - \rho F_{out}(t) \tag{2}$$

For a pure component at constant temperature, the density ρ is constant, and the mass balance can be simplified as follows:

$$\frac{dV(t)}{dt} = F_{in}(t) - F_{out}(t) \tag{3}$$

The volumetric output flow rate F_{out} is a function of the liquid level h and is typically expressed as $F_{out} = f(h)$. In many applications F_{out} is considered proportional to the square root of the liquid level h:

$$F_{out}(t) = c\sqrt{h(t)} \tag{4}$$

where c is a constant. By replacing equation 4 in 3 and computing volume as $V = Ah$, we obtain:

$$\frac{d(Ah(t))}{dt} = F_{in}(t) - c\sqrt{h(t)} \tag{5}$$

Equation 5 can be numerically solved for any $F_{in}(t)$, obtaining the evolution of the liquid level in the storage tank. The volumetric input flow rate $F_{in}(t)$ is an independent variable (from outside the storage tank) that causes changes within the tank. F_{in} is called the input variable.

The liquid level in the storage tank $h(t)$ describes the effect of the input variable on the tank. h is called the output variable. An output variable is a dependent quantity. At the same time, $h(t)$ is the solution to the mathematical model (Equation 5), providing complete information about the state of the storage tank level at any given moment.
h is also a state variable.
A – the cross-sectional area of the storage tank, and the constant c are parameters of the model.

Exercises

Be it a storage tank with a cross-sectional area A = 20 m^2, fed with an input flow rate F_{in} = 2 m^3/h. The tank has a useful volume of 100 m^3, resulting in a maximum level of h_{max} = 5 m. Constant c = 0.8 m^2/h.

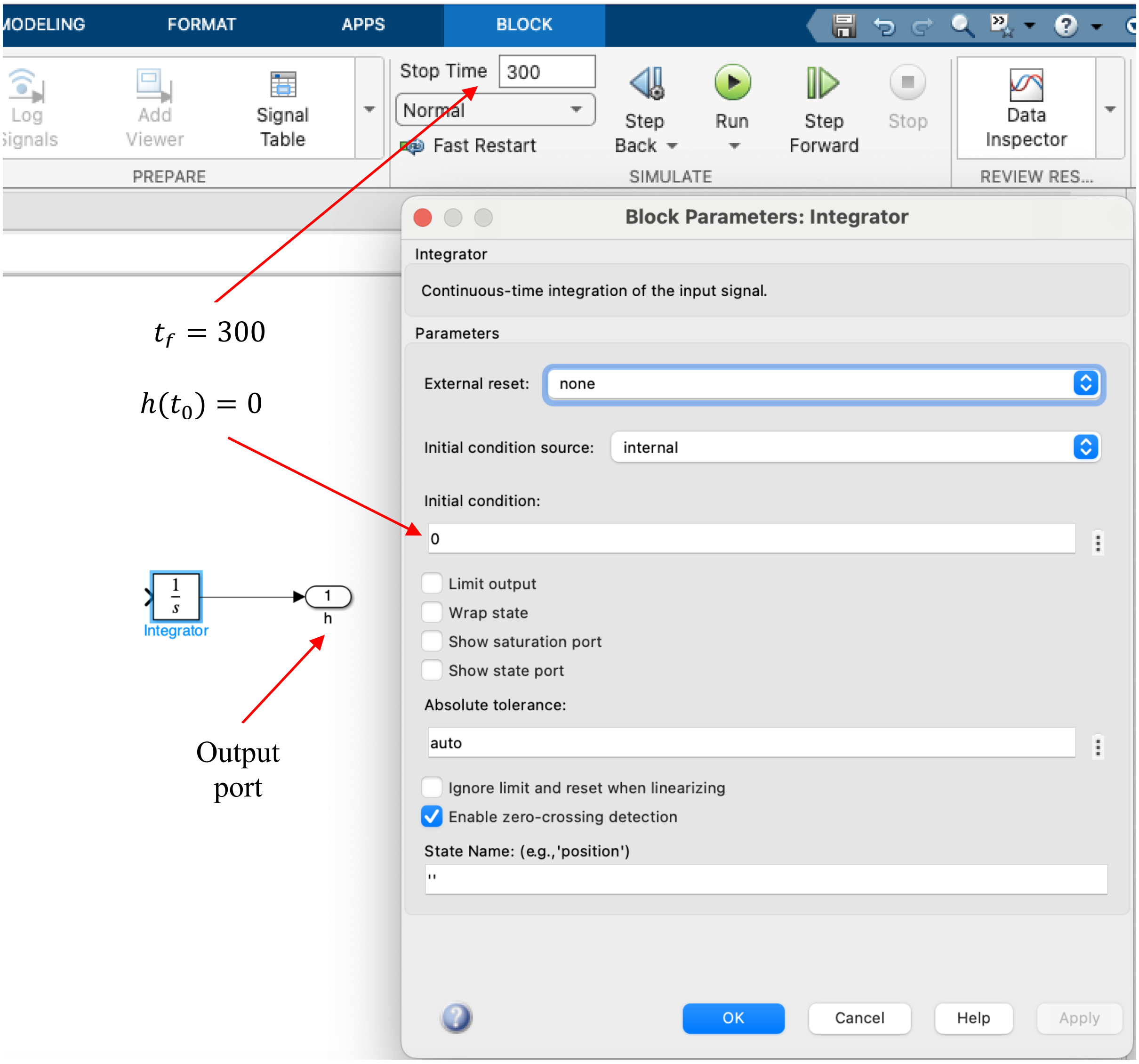
a. Build a Simulink model and simulate the storage tank level h over a period of 300 de ore (t_f = 300). It is assumed that the tank is empty at the start of the supply: $h(t_0) = 0$.
   - Create a script file that calls the Simulink model and graphically represents the tank level h and the output flow rate F_{out}.
   - The constant c can represent a manual valve. Modify its value in the range [0 1] and observe its effect on h and F_{out}.
b. Implement an ON/OFF controller for the storage tank level at h = 5 m, as the supply is controlled by an ON/OFF valve.
   - Add F_{in} at the previous graph.
c. Implement an ON/OFF controller with hysteresis for the storage tank level in the range 4.9 to 5.1 m.

Implementation in Matlab/Simulink

We will implement the following differential equation in Simulink:

$$\frac{dh(t)}{dt} = \frac{F_{in}(t) - c\sqrt{h(t)}}{A}$$

We can start by dragging an Integrator block, and connect its output to an Output Port block. The Output Port block makes the data available in the Workspace. Initial conditions $h(t_0) = 0$ and the final simulation time $t_f = 300$ are input as shown in Figure 2.



Stop Time 300

Normal

Fast Restart

Block Parameters: Integrator

Integrator

Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:

0

☐ Limit output

☐ Wrap state

☐ Show saturation port

☐ Show state port

Absolute tolerance:

auto

☐ Ignore limit and reset when linearizing

☑ Enable zero-crossing detection

State Name: (e.g.,'position')

''

OK Cancel Help Apply

Figure 2. Setting initial conditions and the simulation time

In Figure 3 we find an example of developing the mathematical model by associating blocks such as Divide, Sum, Gain, Constant, Sqrt, and To Workspace.

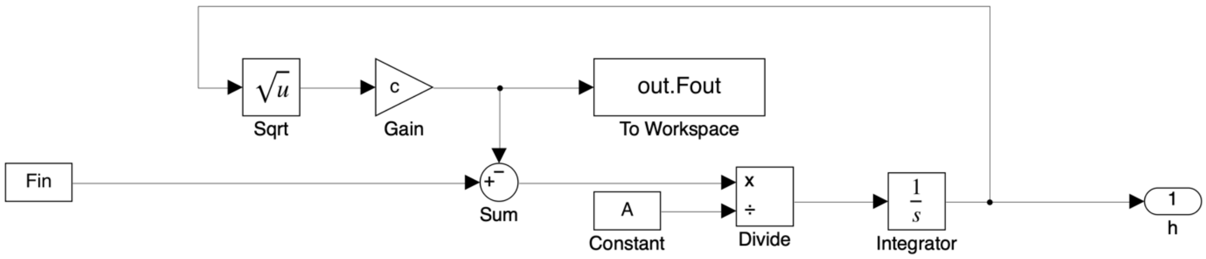


Figure 3. Simulink Model of the Storage Tank Level

This model requires the parameter values of the model. These values will be loaded into the Workspace using a script file:

```
clear; close

%% Model parameters

Fin = 2;    % Input flow rate (m3/h)
c = 0.8;    % constant (m2/h)
A = 20;     % Cross-sectional area (m2)
```

The Simulink model can be called as follows:

```
%% Call Simulink model

StopTime = 300;

SimIn = Simulink.SimulationInput('TancStocare');
SimIn = SimIn.setModelParameter("StopTime",num2str(StopTime));

SimOut = sim(SimIn);
```

TancStocare is the name of the Simulink file. The model was called using the sim function. This can be done by calling the model name (i.e. SimOut = sim('TancStocare')). However, defining a SimulationInput object comes with the advantage that the Simulink model can be modified without the need to open it. In the example above, the simulation time was modified $t_f = 300$.

The SimOut object, which can be found in the Workspace after simulation, is a structure that contains the simulation time (i.e. SimOut.tout) and the level of the storage tank (i.e. SimOut.yout{1}.Values.Data).

For the graphical representation of the volumetric output flow rate, a To Workspace block was added (Figure 3), which makes the data available in the Workspace, in the same SimOut structure (i.e. SimOut.Fout.Data).

Thus, the graphical representation of the storage tank level and the output flow rate can be achieved using the following instruction:

```
%% Graphical representation

figure(1)
subplot(211)
plot(SimOut.tout,SimOut.yout{1}.Values.Data); hold on
plot([0 StopTime],[5 5],'r--'); ylim([0 6])
xlabel('Timp (h)'); ylabel('Nivel (m)'); grid; grid minor
legend('Nivel (m)','Nivel maxim (m)','Location','southeast')
subplot(212)
plot(SimOut.tout,SimOut.Fout.Data); hold on
xlabel('Timp (h)'); ylabel('Debit iesire (m3/h)'); grid; grid minor
```

To simplify the representation of the model, the blocks will be selected as shown in Figure 4. Right-click on the selection, then choose "Create Subsystem from Selection." After creating the subsystem, it will be necessary to rename it, as well as the Input port and Output port blocks inside it.

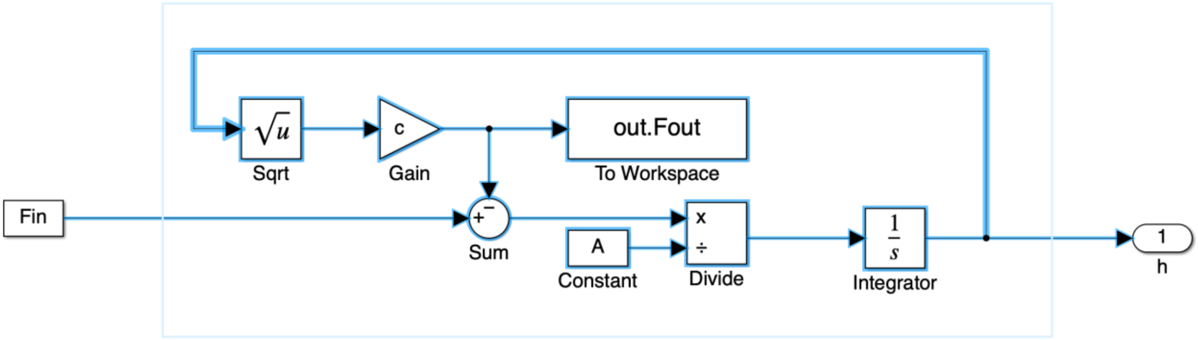
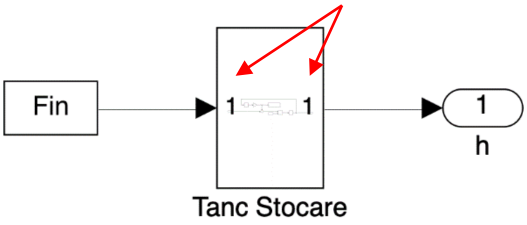


Figure 4. Create a subsystem for the storage tank

The result will be a compact subsystem that contains the mathematical model determining the level of the storage tank.



Redenumire

Figure 5a. Storage tank subsystem

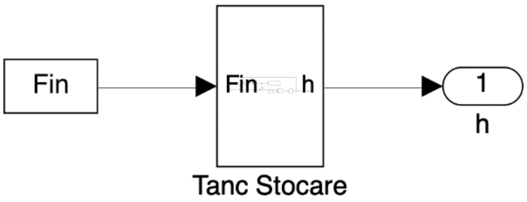


Figura 5b. Subsystem of the storage tank after renaming the input and output

After simulating the model, the following graphical results were obtained in Figure 6. At this input flow rate, F_{in} = 2 m^3/h, the tank level will exceed the maximum allowed value h_{max} = 5 m. Therefore, closed-loop control of the storage tank level is necessary to avoid overflowing.

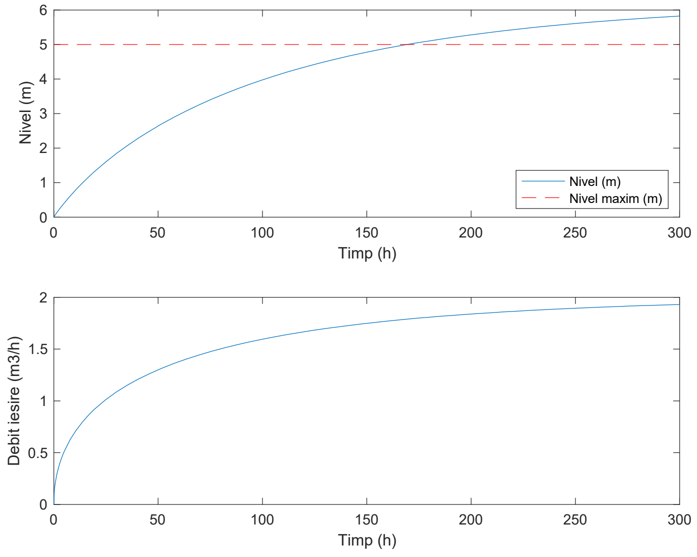


Figure 6. Time evolution of the level and output flow rate

The implementation of the level controller was carried out as seen in Figure 7. A Manual Switch block was added to easily compare simulations with and without the controller. To control the Manual Switch block from the script file, the following instruction will be added when defining the SimulationInput object (1 - Automatic, 0 - Manual).

```
Regulator = 0;      % 1 - with regulator;      0 - without regulator
SimIn = SimIn.setBlockParameter("TancStocare/Manual Switch","sw",num2str(Regulator));
```

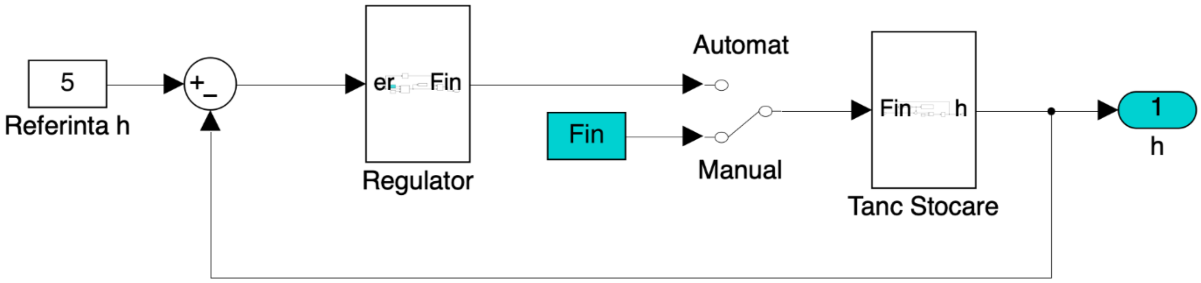


Figure 7. Level controller of the storage tank

The ON/OFF level controller consists of adding a Switch block that allows the value $F_{in} = 2$ to pass if the error is > 0 and $F_{in} = 0$ otherwise. If error messages occur, the solver settings should be accessed, and Adaptive should be selected for Signal Threshold.

A fixed-step integration method can be selected, and the integration step can be specified.

The hysteresis level controller can be implemented using a single Relay block.

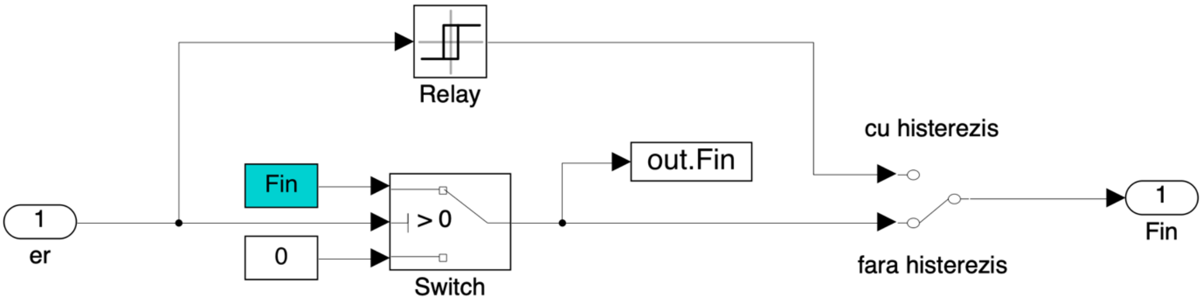


Figure 8. Regulator Subsystem

To facilitate the comparison between the two controllers, a Manual Switch block was added. Additionally, to graphically represent the input flow rate a To Workspace block was used.

```
Histerezis = 0;     % 1 - with histerezis;    0 - without histerezis
SimIn = SimIn.setBlockParameter("TancStocare/Regulator/Manual Switch","sw",num2str(Histerezis));
```

For the graphical representation of the input flow rate F_{in} the following instructions were added:

```
if Regulator == 0
   plot(SimOut.tout,Fin*ones(1,length(SimOut.tout)),'o-')
else
   stairs(SimOut.tout,SimOut.Fin.Data,'o-')
end

legend('Debit iesire (m^3/h)','Debit intrare (m^3/h)','Location','southwest')
```

Application 2: Development of a Dynamic Mass Balance Model for an Aerobic Wastewater Treatment Process

Theoretical aspects

The general dynamic model for continuous processes. A continuous process is one in which fresh culture medium is continuously added to the reactor, and thus, $F_{in} = F_{out} = F$ (figure 1). In a continuous system, the volume is constant, and by substituting $D = F/V$ and dividing by V on both sides of the general model equation (2.18), we obtain:

$$\frac{dc_i}{dt} = r_i + Dc_{i,0} - Dc_i \tag{1}$$

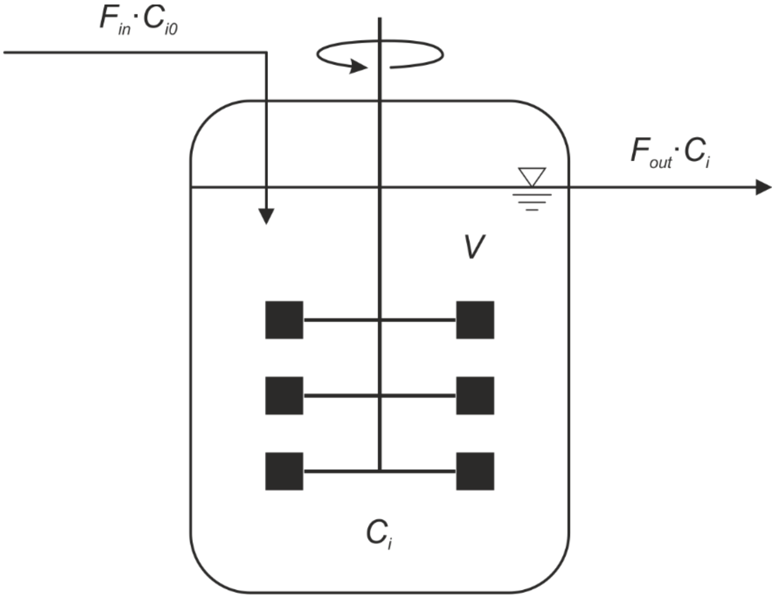


Figure 1. Continuous Stirred-Tank Bioreactor

The general dynamic model 2.18 describes a process that does not involve diffusive fluxes. When it comes to biotechnological processes, the presence of a diffusive flux is justified only for gases that can dissolve in water and participate in the process. The options are quite limited, most commonly modeling the dynamics of dissolved oxygen, and sometimes that of dissolved carbon dioxide. When modeling dissolved gases, the general dynamic model is enriched with a term describing the gas-liquid mass transfer rate (equation 2.15). If we rewrite the model for a continuous process taking place in a fully mixed reactor with a single convective flux, equation 1 will become:

$$\frac{dc_i}{dt} = r_i + Dc_{i,0} - Dc_i + N_i \tag{2}$$

The term describing the gas-liquid mass transfer rate, N_i, will always be positive (or 0) because mass is transferred from gas to liquid.

Dynamic Mass Balance Model

The mass balance model combines terms describing kinetic rates (growth of activated sludge and consumption of substrate and dissolved oxygen) and transport terms and has the following form (Ifrim, 2012):

$$\frac{dX_{\mathrm{B}}}{dt} = r_{X_{\mathrm{B}}} - DX_{\mathrm{B}} + rDX_{\mathrm{R}} - rDX_{\mathrm{B}} \tag{3}$$

$$\frac{dS_{\mathrm{S}}}{dt} = r_{S_{\mathrm{S}}} + DS_{\mathrm{S,in}} - DS_{\mathrm{S}} \tag{4}$$

$$\frac{dS_{\mathrm{O}}}{dt} = r_{S_{\mathrm{O}}} \cdot 10^3 + DS_{\mathrm{O,in}} - DS_{\mathrm{O}} + N_{O_2} \tag{5}$$

$$\frac{dX_{\mathrm{R}}}{dt} = D_s X_{\mathrm{B}} + rD_s X_{\mathrm{B}} - rD_s X_{\mathrm{R}} - \beta D_s X_{\mathrm{R}} - \eta D_s (1-\beta) X_{\mathrm{R}} \tag{6}$$

where D – is the aerobic basin dilution, D_s – is the clarifier sludge dilution, r – is the recirculation rate, and β – is the excess sludge rate, they have the following expressions:

$$D = \frac{F_f}{V}; \qquad D_s = \frac{F_f}{V_s} = D\frac{V}{V_s}; \qquad r = \frac{F_r}{F_f}; \qquad \beta = \frac{F_e}{F_f};$$

The volumetric rates of activated sludge growth, $r_{X_{\mathrm{B}}}$, substrate consumption, $r_{S_{\mathrm{S}}}$, and dissolved oxygen consumption, $r_{S_{\mathrm{O}}}$, are given by equations 3.4, 3.5, and 3.6, respectively. The gas-liquid mass transfer rate of oxygen, N_{O_2}, is calculated with equations 3.7 and 3.8. r is the recirculation coefficient, representing the ratio of the recirculation flow rate to the influent flow rate, and β is the excess sludge coefficient representing the ratio of the excess flow rate to the influent flow rate. $S_{\mathrm{S,in}}$ and $S_{\mathrm{O,in}}$ represent the concentrations of organic pollutants (substrate) and dissolved oxygen in the influent.

Exercises:

build a Simulink model and simulate the evolution of all state variables over a period of 300 hours ($t_f = 300$). The initial conditions are: $x(t_0) = [X_B(t_0) \quad S_S(t_0) \quad S_O(t_0) \quad X_R(t_0)] = [0.5 \quad 0.8 \quad 2 \quad 0]$.

create a script file that calls the Simulink model and graphically represents all 4 state variables.

Implementation in Matlab/Simulink:

We can start by constructing the subsystem for the specific growth rate. An example implementation can be found in Figure 1..

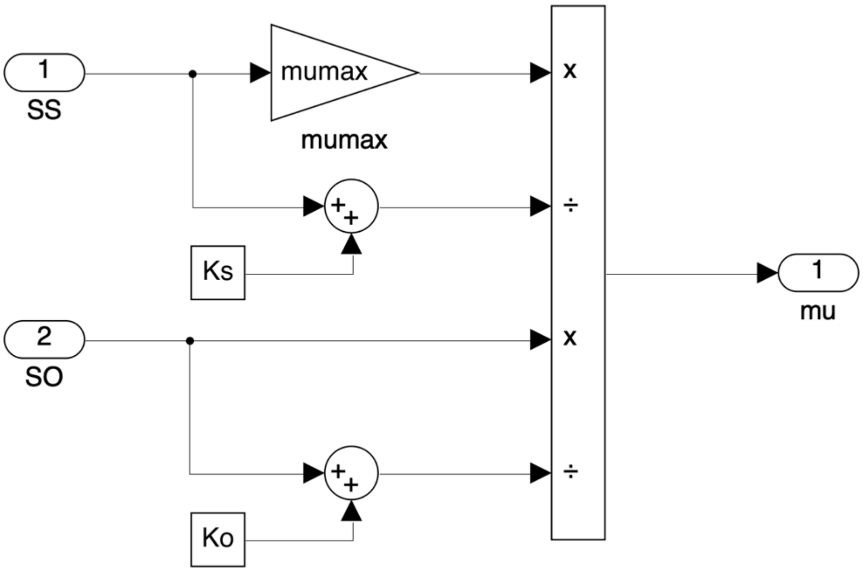


Figure 1. Specific Growth Rate Subsystem

This subsystem will be used for constructing the model for the aerobic basin. In Figure 2, we find an example of the implementation of the three state variables that characterize the aerobic basin: X_B, S_S și S_O.

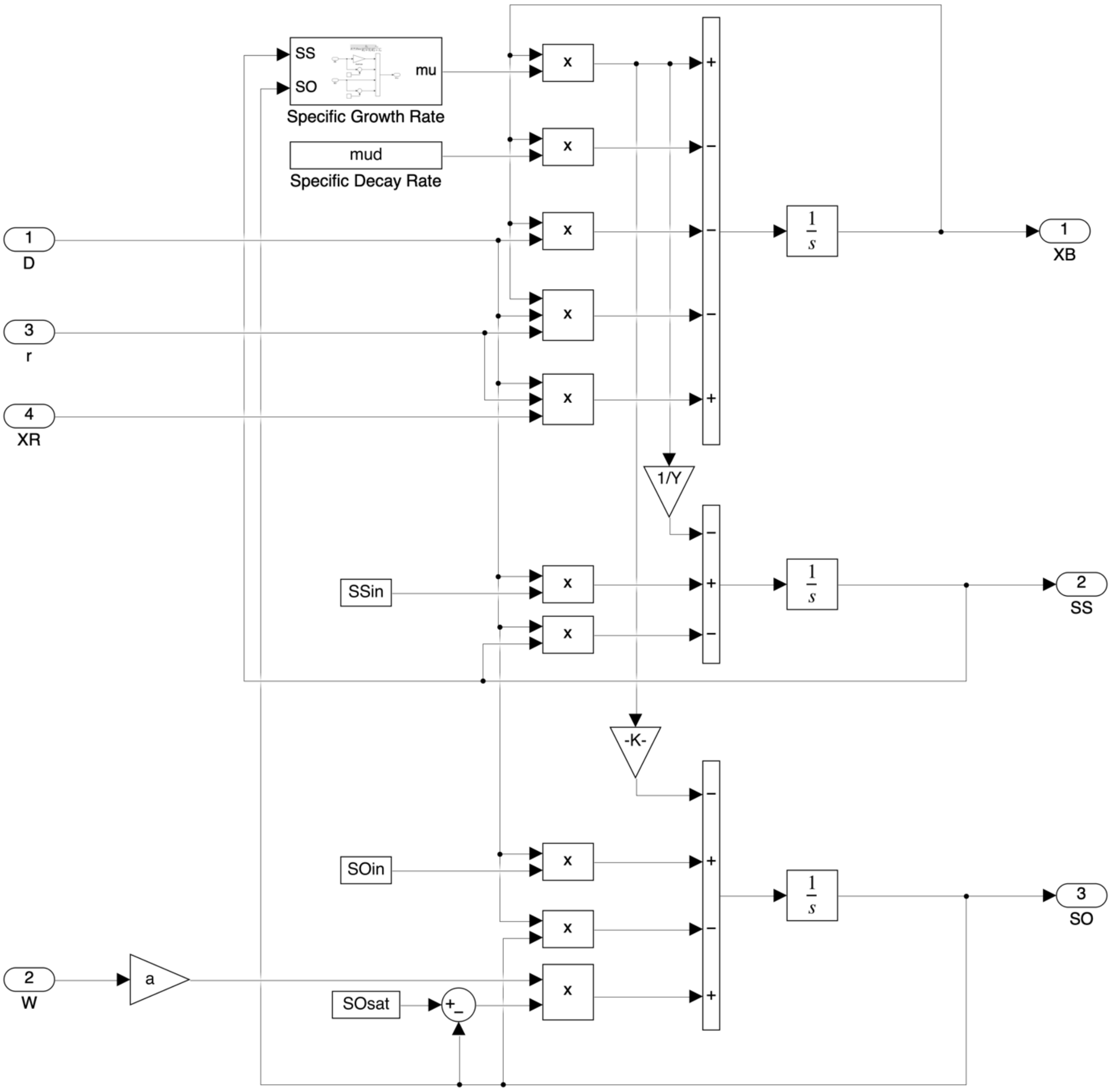


Figure 2. Aerobic Basin Subsystem

The clarifier will be implemented in a separate subsystem, as can be seen in Figure 3.

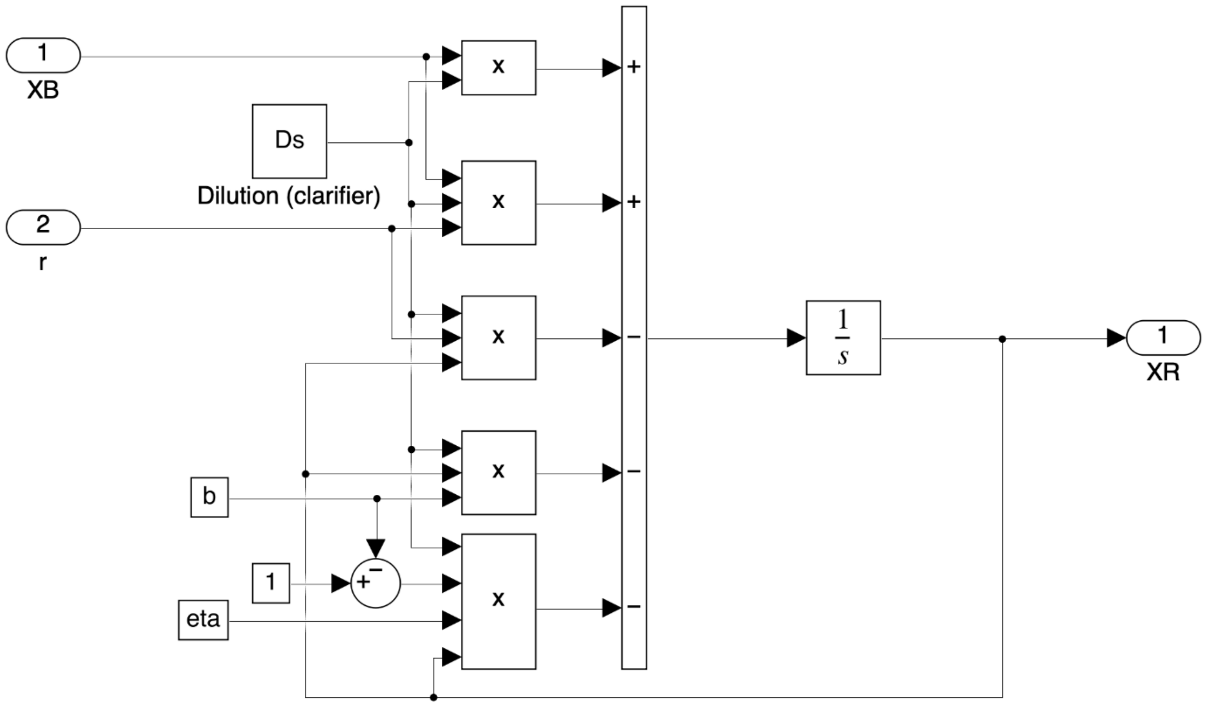


Figure 3. Clarifier Subsystem

The two subsystems will be able to be coupled, as can be seen in Figure 4.

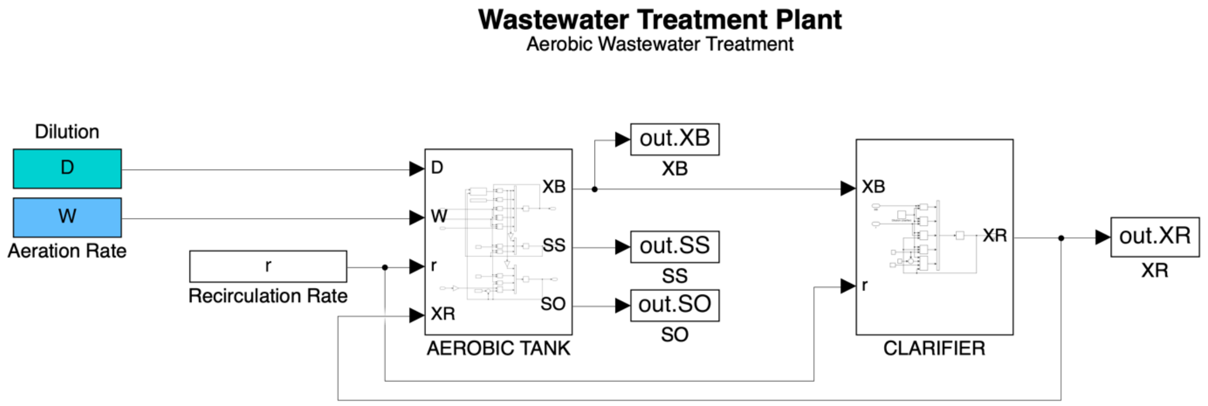


Figure 4. Mathematical Model of an Aerobic Wastewater Treatment Process

The parameters of the model and input variables will be written in a script file.

```
%% Model parameters

mumax = 0.21;   % Maximum specific growth rate [h-1]
mud = 0.02;     % Specific decline rate [h-1]
Ks = 0.18;      % Semisaturation constant [g.L-1]
Ko = 0.2;       % Saturation constant for OD (Oxygen Demand) [mg.L-1]
Y = 0.67;       % Substrate conversion yield [-]

a = 0.0033;     % Diffusion coefficient
SOsat = 8;      % OD concentration at saturation [mg.L-1]

r = 1;          % Ratio between Fr and Ff [-]
```

```
b = 0.2;        % Ratio between Fe and Ff  [-]
V = 35;         % Volume of the aerobic basin [m3]
Vs = 6;         % Volume of the sludge layer [m3]
eta = 0.25;

%% Input Variables

D = 0.03;       % Dilution of the aerobic basin [h-1]
Ds = D*V/Vs;    % Dilution of the sludge layer [h-1]

W = 2100;          % Aeration rate [l.min-1]

SSin = 4;       % Influent substrate concentration [g.L-1]
SOin = 2;       % Influent OD concentration [g.L-1]
```

Calling the Simulink Model

```
%% Call the Simulink model

StopTime = 300;

SimIn = Simulink.SimulationInput('Apauzata');
SimIn = SimIn.setModelParameter("StopTime",num2str(StopTime));

SimOut = sim(SimIn);                                          % Simulates the model
```

Graphical representation

```
%% Graphical representation

figure(1)
subplot(221)
plot(SimOut.tout,SimOut.XB.Data)
xlabel('Time (Hours)'); ylabel('Namol activat (g/L)')
grid; grid minor; hold on
subplot(222)
plot(SimOut.tout,SimOut.SS.Data)
xlabel('Time (Hours)'); ylabel('Substrat (g/L)')
grid; grid minor; hold on
subplot(223)
plot(SimOut.tout,SimOut.SO.Data)
xlabel('Time (Hours)'); ylabel('Oxigen dizolvat (mg/L)')
grid; grid minor; hold on
subplot(224)
plot(SimOut.tout,SimOut.XR.Data)
xlabel('Time (Hours)'); ylabel('Namol recirculat (g/L)')
grid; grid minor; hold on
```

After simulating the model we obtain:

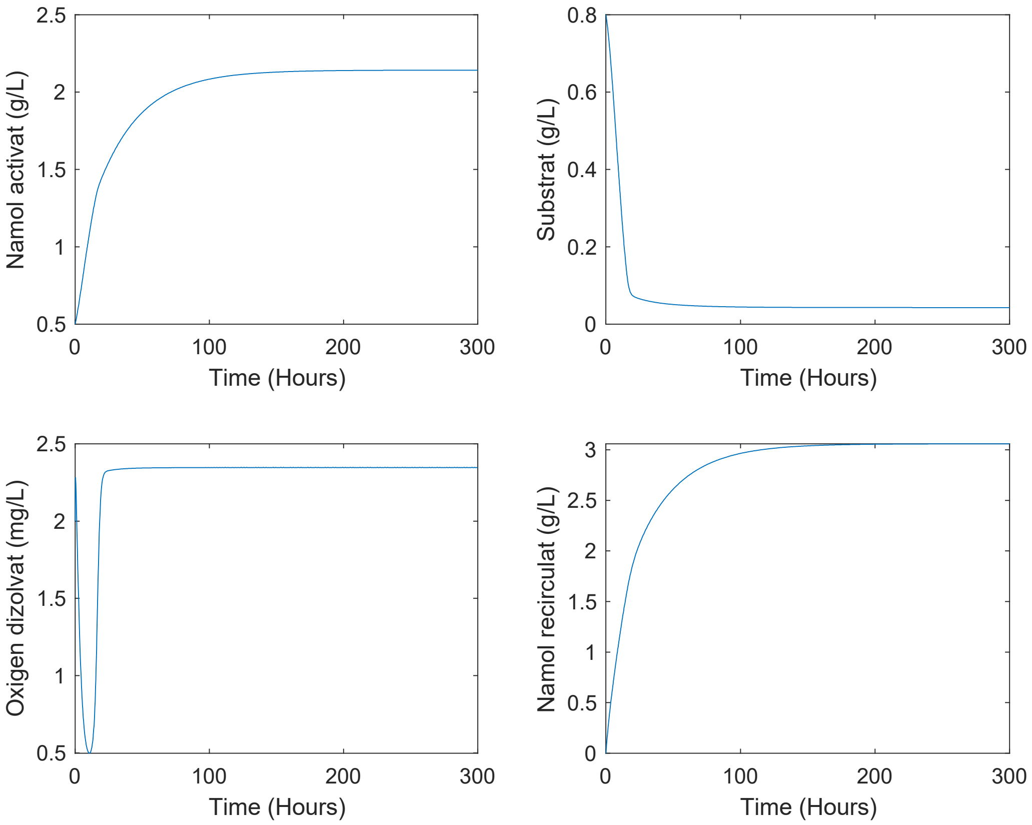


Figure 5. The results obtained after simulation

Application 3: Dissolved Oxygen Control in an Aerobic Wastewater Treatment Process

Dissolved oxygen control is crucial for any aerobic process, especially in the case of aerobic biological wastewater treatment. Dissolved oxygen, S_O is measured using electrochemical or optical sensors and is controlled by the aeration rate, W. The dissolved oxygen loop must exist regardless of any other type of control implemented in the facility. Effective dissolved oxygen control can lead to an efficiency in the removal of organic pollutants of up to 10%.

Exercises:

Implement a closed-loop control system for dissolved oxygen using the mathematical model of the aerobic wastewater treatment process developed in the previous application.

- Implement a PI controller tuned using the "trial and error" method. The reference value for dissolved oxygen is $S_O = 2$ mg/L.
- Graphically represent the aeration rate and dissolved oxygen.

Implementation in Matlab/Simulink

The first step will be to incorporate the entire model of the aerobic treatment station into a subsystem with the aeration rate W as input and dissolved oxygen S_O as output.

Around this subsystem, the closed-loop control scheme will be built, as seen in Figure 1.

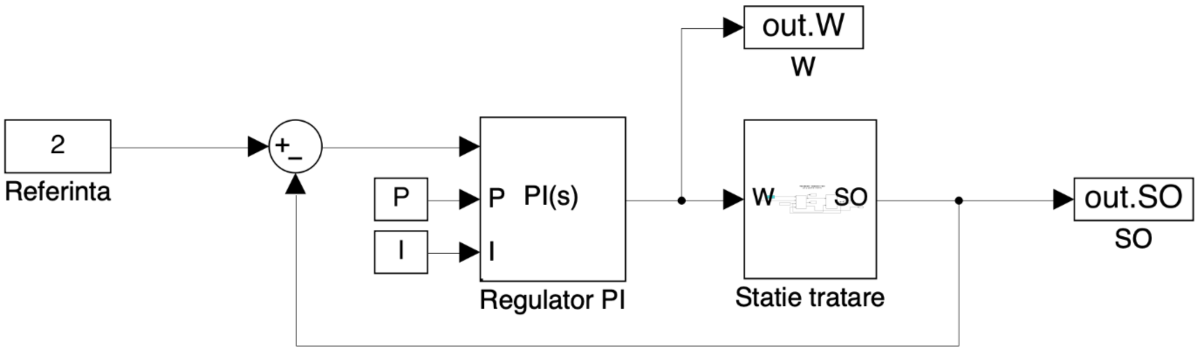


Figure 1. Closed-Loop Control Scheme for Dissolved Oxygen

The parameters of the PI controller have been added to the script file.

```
%% Regulator paramters

P = 1;
I = 1;
```

For the graphical representation of the input variable W, a*To Workspace*, block has been added. The following instructions have been added to the script file:

```
figure(2)
subplot(211)
plot(SimOut.tout,SimOut.W.Data)
xlabel('Time (Hours)'); ylabel('Rata de aerare (L/h)')
grid; grid minor; hold on
subplot(212)
plot(SimOut.tout,SimOut.SO.Data)
xlabel('Time (Hours)'); ylabel('Oxigen dizolvat (mg/L)')
grid; grid minor; hold on
```

Bibliography

Dochain D. Automatic Control of Bioprocesses. John Wiley & Sons, Inc. Great Britain, 2008.

Doran MP. Bioprocess Engineering Principles. Academic Press, 1995.

Gray NF. Water Technology. An Introduction for Environmental Scientists and Engineers. Second Edition. Elsevier, Oxford. 2005.

Ifrim GA. Comanda proceselor de interes pentru mediu (tratarea biologică a apelor uzate și creșterea microalgelor în fotobioreactor). Teza de doctorat. Universitatea „Dunărea de Jos” din Galați, 2012.

Kravaris, C., & Kookos, I. (2021). Understanding Process Dynamics and Control (Cambridge Series in Chemical Engineering). Cambridge: Cambridge University Press. doi:10.1017/9781139565080

Sablani SS, Shafiur R, Datta AK and Majumdar AS. Handbook of Food and Bioprocess Modeling Techniques. Taylor & Francis, USA, 2006.

Snape JB, Dunn IJ, Ingham J, Prenosil JE. Dynamics of Environmental Bioprocesses. Modeling and Simulation. VCH, 1995.

Cybersecurity

- course notes –

I. Introduction

Since the global network of digital communications has become an integral part of our lives on all fronts, cybersecurity is essential for protecting data, society, the state, and ensuring economic stability. Cybersecurity refers to the ability to protect or defend the cyberspace from cyber attacks. In other words, it involves the ongoing effort to protect individuals, organizations, and governments from digital attacks targeting unauthorized use or destruction of critical systems, networks, software applications, and sensitive data.

It should be emphasized that the protection of identity, data, and electronic devices is important both personally and professionally. At the organizational level, it is the responsibility of all employees and partners to ensure the preservation of the reputation, protect the data, and safeguard the customers of the company they work for. At the state level, national security, the safety, and well-being of all citizens are at stake.

It is a well-known fact that cybersecurity threats are on the rise, and cybercriminals seek to exploit any vulnerability they can find to steal information or money. Cybersecurity issues and attacks on critical infrastructure are among the top 5 global risks according to the World Economic Forum Global Risk Report (2023).

Cyberattacks increase with every new digital connection made in the world. Therefore, cybersecurity professionals who can protect and defend an organization's network are in high demand at the moment. Among the most sought-after and well-paying jobs in cybersecurity are Chief Information Security Officer (CISO: $108,000 - $233,000), Information Security Manager ($82,000 - $156,000), Security Architect ($87,000 - $158,000), Network Security Engineer ($63,000 - $140,000), and others.

II. Cybersecurity purposes

The goals of cybersecurity can be summarized in the following key principles, encompassed in the information security guide known as the CIA Triad (Confidentiality, Integrity, Availability):

- **Confidentiality** – Prevents the disclosure of information to unauthorized users or processes, including means of protecting confidential and proprietary information. Various methods and technologies can be employed to ensure data confidentiality and enhance privacy, such as data encryption, access control, anonymization, and tokenization.
- **Integrity** – Prevents unauthorized modification or destruction of information, focusing on authenticity, accuracy, consistency, and trust in devices, networks, applications, information, and data throughout their life cycle. Methods used to ensure data integrity include hashing, data validation checks, consistency checks, and access control. Data integrity systems may incorporate one or more of these methods.
- **Availability** – Ensures that systems, networks, information, and data are available when needed by authorized users or processes. Methods used to ensure availability include system redundancy, system backups, increased system resilience, equipment maintenance, operating system and application updates, and proactive plans for recovery after unforeseen disasters.

Data protection must be ensured throughout their life cycle:

- At Rest: when data is stored, and no user or process is accessing, requesting, or modifying it.
- In Transit: when data is transmitted between network-connected devices.
- During Processing: when data is input, manually or automatically modified, processed, or displayed.

These principles provide a comprehensive framework for establishing robust cybersecurity measures and are critical for safeguarding information assets in the digital age.

III. Categories of cybercriminals

Security threats are diverse and can be categorized in various ways. This classification allows organizations to assess and estimate the potential impact of these threats, prioritize them, and organize defenses against them.

The list of security threats includes: Software Attacks: such as Distributed Denial of Service (DDoS) attacks or computer viruses, Software Bugs: errors that allow the unauthorized sharing of files or offline functioning of applications, Hardware Errors and Failures: malfunctions in hardware equipment, Human Errors: mistakes made during data input, manipulation of equipment, or the disclosure of confidential information, Sabotage: deliberate acts to disrupt operations, Information Theft: stealing information or equipment from unsupervised locations, Utility Disruptions: interruptions in utility supplies, such as power outages, Natural Disasters: events like floods, fires, earthquakes, or severe weather conditions (storms, hurricanes, tornadoes).

Threat sources can be both internal and external. Internal threats may cause damage surpassing that caused by external threats due to the potential for direct access to the organization's devices and network. Additionally, attackers with internal knowledge possess information about resources, confidential data, and security measures implemented within the organization.

Internal threats can originate from employees and other contractual individuals, both current and former, who misuse the organization's resources and data. They may accidentally or intentionally destroy or alter them, misconfigure equipment or applications, affecting server operations, infrastructure, network equipment, connecting infected storage media, or accessing malicious emails or websites.

External threats usually come from attackers or groups of attackers, ranging from amateurs to highly skilled individuals. They have the capability and tools to exploit vulnerabilities in the organization's equipment, network, and software applications or gain access to them using social engineering techniques.

Advanced Persistent Threats (APTs) represent continuous, complex attacks that employ sophisticated espionage tactics, advanced malware, and involve multiple attackers to gain access, analyze, and exploit the target's network, devices, and data. APTs are typically well-orchestrated, well-funded, remain undetected for long periods, and have devastating consequences.

In Figure 1, the main types of cybercriminals are presented, grouped into categories based on the source of threats and the level of expertise. Amateur criminals, also known as script kiddies, possess few skills and use tools or instructions found on the internet to exploit network vulnerabilities for financial or personal gain. Even though the level of their attacks is not always highly sophisticated, the results can be devastating.

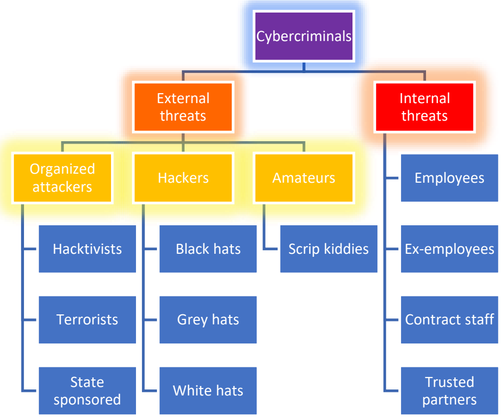


Fig. 1 Types of cybercriminals

Another category of attackers – hackers – are well-trained and informed, targeting networks or computers by exploiting their vulnerabilities or using social engineering techniques, acting individually or organized in groups and having different objectives:

- White hat hackers – penetrate networks and systems with the owner's permission to improve their security; also known as ethical hackers.
- Gray hat hackers – behave like white hat or black hat hackers depending on their interests; they either report identified vulnerabilities to system owners or publish them on the internet for exploitation by other attackers.
- Black hat hackers – exploit vulnerabilities for personal, financial, or political gains.
- Organized hackers
    - Cybercriminals – are groups of professional criminals whose goal is to gain control, power, and money.
    - Hacktivists – aim to raise public awareness about certain issues they consider important.
    - Terrorists – criminals who use technology to carry out premeditated attacks with the purpose of spreading fear and causing physical, social, political, or economic disruptions.
    - State-sponsored groups – conduct cyber attacks for the benefit of the governments that financially and logistically support them.

The Internet has also become an arena for conflicts between nations, known as cyberwarfare. The battles in cyberspace involve infiltrating the information systems and networks of other nations to sabotage and cause damage, disrupt services, and engage in industrial or military espionage. Cyber warfare can destabilize nations, impact trade relations, erode the trust of the population in the government and state authorities, all without the need for a military invasion.

IV. Categories of cyberattacks

The majority of cyber attacks unfold on multiple fronts, combining various techniques to compromise and exploit a system, network, or application. Attackers exploit software and hardware vulnerabilities, use various types of malicious software (malware), and rely on diverse methods of infiltration, deception, manipulation, and fraud based on social engineering.

Some of the most common types of cyber attacks include:

- Malware Usage: Malicious software that can be used to steal or destroy data, bypass access control measures, or compromise/damage a device, network, or application.
- Man-in-the-Middle Attacks: Intercept, modify, and transmit false information between two devices with the goal of stealing data or impersonating one of the devices.
- Zero-Day Attacks: Exploit software vulnerabilities in applications before they become publicly known and patched.
- Spoofing Attacks: Falsify MAC or IP addresses to disguise the attacker's device as a valid one.
- Flooding Attacks: Compromise network intermediary equipment by flooding it with false MAC addresses.
- Denial of Service (DoS) Attacks: Send large volumes of data at a rate the target system or network cannot handle, resulting in slowdowns or blockages.
- Deception Techniques: Manipulate or deceive users to make them perform specific actions or disclose sensitive information.

4.1. Types of malware

The main types of malware used to steal data, bypass access control mechanisms, cause damage, or compromise systems and networks are as follows:

- Virus: Executable code that replicates and attaches itself to other executable files to modify, steal, or delete data. They spread through storage devices (USB, CD, DVD), network resource sharing, or email.
- Worm: Code that does not need a host file to be executed, exploiting network vulnerabilities to replicate and spread rapidly, leading to network slowdown or blockage.
- Trojan: Malware that performs malicious operations disguised under the guise of valid actions, attaching to non-executable files and exploiting user privileges.
- Ransomware: Malware designed to usually block the operation of a computer system or data stored on it through encryption. It threatens to maintain the blockage or erase data until a payment is made to the attacker. Payment does not guarantee unlocking or decrypting the data.
- Spyware: Acts as part of legitimate software or a Trojan, bypassing security systems to track and spy on the user.
- Adware: Malware designed to deliver advertisements automatically, potentially accompanied by spyware.
- Scareware: Software that exploits the user's fear, leading them to take actions that allow the installation of other types of malware.

Regardless of the type of malware that has infected a computer system or network, common symptoms can be recognized, such as increased processor usage, decreased work speed, system blocking or shutdown, modification, deletion, or encryption of files, presence of unknown files or applications, execution of unknown processes, and sending messages without the user's consent.

4.2. Types of deception

One of the most effective and sometimes simpler ways attackers can obtain information about computer systems and an organization's network is through deception, manipulating the victim to perform certain actions (social engineering). This involves observing the victim's actions while entering PIN codes or passwords, either up close or with the help of various remote viewing devices (shoulder surfing), searching for information in discarded, non-shredded documents in the trash (dumpster diving), pretending to be someone else (impersonating), or misleading the victim by creating fear and inducing irrational behavior (hoax).

Regardless of how robust security measures are, how accurately network equipment, servers, and systems are configured, and how restrictive access control techniques may be, deception techniques cannot be prevented without proper awareness and training of users.

4.3. Types and Tactics of Social Engineering

Social engineering exploits human nature, relying on people's sociability, their willingness to help, the implicit trust people have in each other, or takes advantage of their weaknesses to manipulate them and make them perform certain actions or disclose sensitive, confidential information.

There are several types of social engineering attacks, including:

- Pretexting: The attacker lies, pretending to need personal, medical, or financial data to confirm the target's identity.
- Quid Pro Quo: Information is requested in exchange for something else, such as a gift or a free vacation.
- Identity Fraud: Stolen identity is used to obtain goods or services.

Tactics used by attackers to gain access to sensitive information include the use of authority, intimidation, consensus building in actions, highlighting a limitation in availability, imposing a time limit to emphasize urgency, and building a rapport of familiarity or trust with the victim..

V. Cyber Attack Lifecycle

A cyber attack unfolds in several stages. Lockheed Martin has developed a framework model for identifying and preventing cyber attacks, known as the ***Cyber Kill Chain***[1] (or the lifecycle of a cyber attack). It comprises the following seven steps, grouped into three stages:

- **Before compromise**
  1. Reconnaissance - Identify and select the target, gather as much information as possible from the public domain and social media (e.g., employees' email addresses, their roles, hobbies, etc.) or by using various tools for scanning network vulnerabilities, devices, services, and applications: network analyzers (packet or protocol scanners, e.g., tcpdump and Wireshark), port scanners (e.g., Nmap), vulnerability scanners (e.g., Nessus), password crackers (e.g., John the Ripper), etc.

[1] Cyber Kill Chain® | Lockheed Martin (Septembrie, 2023)

2. Weaponization - Determine the method to compromise the target and identify or create exploits that are embedded in a malicious payload
3. Delivery – Send the malicious package to the victim via email, web redirection, USB, infected file sharing, or other methods.

- **During compromise**

4. Exploitation – Exploit a vulnerability to execute code on the victim's system either by the user's involuntary triggering of the exploit (e.g., clicking on a link or opening an infected file attached to an email) or by the remote triggering by the attacker of an exploit of a known vulnerability.
5. Installation – Install malware programs and backdoors on the compromised system.

- **After compromise**

6. Command and control – Control the victim's device remotely through an encrypted command and control channel or a C2 server (command & control). To maintain undetected C2 traffic, encryption is used, and methods such as proxy avoidance, remote access tools, port hopping, tunneling, etc., are employed.
7. Actions on the objective – The attacker steals information, destroys or modifies critical systems, networks, or data, attacks other devices in the network, or uses the compromised target as a platform to support an attack on another victim.

It is also noteworthy that the first four steps constitute the unauthorized access phase, while the last four steps constitute the unauthorized use phase..

VI. Security countermeasures

The security countermeasures available to an organization fall into two major categories:

- **Technology-Based Countermeasures:** These include the use of protective devices (firewalls) both in hardware and software, intrusion detection and prevention equipment (IDS and IPS), and various types of network, port, and vulnerability scanners.
- **Human Factor-Based Countermeasures:** This category encompasses the education and periodic training of personnel to raise awareness of cybersecurity issues, as well as the implementation of policies, procedures, guidelines, and best security practices.

To ensure the protection of stored data, security measures include, among others, encrypting sensitive data for confidentiality, creating backups, and recovering altered data to ensure availability. Data in transit is protected using security measures such as encryption to guarantee confidentiality, hashing for authenticity and integrity, and ensuring redundancy to support availability. For data that is neither stored nor in transit, security measures are directed towards better design of applications and software and hardware systems. This ensures that input data is validated, software errors (bugs) are minimized, and output devices are correctly configured..

6.1. Access control

Access control is the process by which access rights are restricted to control the use of the network, equipment, data, services, and applications. It encompasses activities ranging from managing physical access to a location or resource to determining who has access and what they can do with it. Many security vulnerabilities are created by the improper use of access control processes. There are several types of access control: physical control, logical control, and administrative control.

Physical access control involves raised barriers, either outside or within the protected perimeter, to prevent direct physical contact with equipment and the network. The goal is to prevent unauthorized users from gaining physical access to the organization's facilities, equipment, and other assets. Physical control determines who can enter (or exit), where they can enter (or exit), and when they can enter (or exit).

Examples of physical access control include security guards, fences, gates, motion detectors, equipment locks, door locks, access cards, video cameras, specialized entry systems (e.g., mantraps), intrusion detection alarms, etc.

Logical access control refers to hardware and software solutions used to manage access to resources and systems. These technology-based solutions include the tools and protocols that computer systems use for identification, authentication, authorization, and auditing.

For logical access control, encryption, smart cards, passwords, biometric identifiers, access control lists (ACLs), protocols, firewalls, routers, intrusion detection systems (IDS), etc., can be used.

Administrative access control is represented by policies and procedures defined to implement and enforce all aspects of unauthorized access control. Examples of administrative control include policies (objectives, rules, and requirements), procedures (detailed steps necessary to perform an activity), hiring practices (steps an organization takes to find qualified employees), background checks (information from former employers, credit history, criminal history, etc.), data classification (based on sensitivity), training (educating employees on security policies), etc..

6.2. Security services

The concept of administrative access control involves three essential security services: authentication, authorization, and accounting. These services provide the primary framework for access control, preventing unauthorized access to systems, networks, data, or other resources.

Authentication is the process of confirming the identity of a user or system to prevent unauthorized access. It ensures that the person or entity attempting to access resources or perform actions is who they claim to be.

Authorization is the process of granting or denying access to specific resources. Authorization services determine which resources users can access, along with the operations they can perform on those resources. Some systems accomplish this through Access Control Lists (ACLs), which determine whether a user can have certain privileges after authentication.

Authorization can also control when a user has access to a particular resource. For example, employees may have access to a database during working hours, but access is not allowed after the end of the workday.

Accounting is the process of recording and monitoring activities and events occurring on a network or computer system. It keeps track of the operations performed by users: what resources they access, how long, and what changes they make. The purpose of auditing is to ensure compliance with security policies and facilitate the identification of potential threats and abuses, aiming to prevent and detect security incidents..

6.3. Multi-Factor Authentication

A unique identifier ensures the proper association between permitted activities and subjects. For authentication, users provide their identity with a username or ID. A username is the most common method used to identify a user through a combination of alphanumeric characters. A unique identifier ensures that a system can identify each individual user, allowing an authorized user to perform appropriate actions on certain resources.

Additionally, users must prove their identity by providing:

- Something they know – e.g., a password, a phrase, or a PIN code.
- Something they have – a token, a card, or a key.
- Something they are – biometric identifiers – a fingerprint or other physical (face, hand, retina, ear) or behavioral (gesture, voice, gait) characteristic.

In the case of two-factor authentication (2FA) or multi-factor authentication (MFA), to verify someone's identity, a combination of at least two of the authentication means mentioned above is required.

For example, a bank's website may ask for a password and a PIN code that the user receives on their phone. In this case, the first factor is the password, and the second factor is a temporary code, as it proves that you have access to what is registered as the phone. Withdrawing money from an ATM is another simple example of multi-factor authentication because the user needs to have the bank card and know the PIN.

Multi-factor authentication can reduce identity theft incidents because merely knowing a password will not give a cybercriminal direct access to the user's account..

6.4. Cryptography

Cryptography is the science of creating and breaking secret codes. Cryptography is the process of transforming data so that unauthorized individuals cannot easily read it. This process converts the readable message (plaintext) into an encrypted message (ciphertext), which is unreadable, a disguised message. Decryption reverses the process. By storing and transmitting encrypted data, so that only the intended recipient can read or process the data, data protection is provided. This means that unauthorized users cannot easily read sensitive information.

Cryptography requires a key that plays a critical role in encrypting and decrypting the message. The person possessing the key can decrypt the encrypted message, transforming ciphertext into plaintext.

6.4.1. Types of cryptography

The most commonly used types of cryptography involve block ciphers or stream ciphers. Each method differs in how it encrypts groups of data bits.

Block ciphers transform a fixed-length block of plaintext into a corresponding block of ciphertext of 64 or 128 bits. The block size is the amount of data encrypted at any given time. To decrypt this ciphertext, the inverse transformation is applied to the ciphertext block, using the same secret key.

Block ciphers usually result in output data that is larger than the input data because the ciphertext must be a multiple of the block size. For example, the Data Encryption Standard (DES) is a symmetric algorithm that encrypts blocks in 64-bit pieces using a 56-bit key. To achieve this, the algorithm takes one block at a time—e.g., 8 bits at a time—until the entire block is filled. If there is less input data than a full block, the algorithm adds artificial data or spaces until it fills 64 bits.

Stream ciphers encrypt plaintext one bit at a time, sequentially, one bit at a time. Think of a stream cipher as a block cipher with a block size of 1 bit. With a stream cipher, the transformations of these small units of plaintext vary, depending on when they occur during the encryption process. Stream ciphers can be much faster than block ciphers and generally do not increase the size of the encrypted message, as they can encrypt an arbitrary number of bits.

For example, A5 is a stream cipher that provides voice privacy and encrypts mobile phone communications. It is also possible to use DES in stream cipher mode.

Clearly, complex cryptographic systems can combine blocks and streams in the same process.

6.4.2. Classes of encryption algorithms

Modern cryptography uses secure algorithms to ensure that cybercriminals and other malicious actors cannot easily compromise protected information. All encryption methods use keys to encrypt and decrypt messages, and the security of encryption relies on the secrecy of the keys, not the secrecy of the algorithm. In modern cryptography, algorithms are public. Cryptographic keys must ensure the secrecy of the data.

An encryption algorithm is only as good as the key it uses. The more complexity involved, the more secure the algorithm. Key management is an important and highly challenging part of a cryptographic system.

There are two classes of encryption algorithms::

- **Symmetric Encryption** – These algorithms use the same pre-shared key, sometimes called a secret key, to encrypt and decrypt data. Both the sender and the recipient know the pre-shared key before starting encrypted communication. Symmetric algorithms use the same key to encrypt and decrypt the message at both ends of the process. Encryption algorithms using a common key are simpler and require less computational power because they rely on simple mathematical operations.
  Examples:
  - **3DES** (triple DES) - encrypts data three times using DES and uses different keys for at least one of the three passes, resulting in a cumulative key length from 112 to 168 bits.
  - **IDEA** (*International Data Encryption Algorithm*) uses 64-bit and 128-bit blocks, performing 8 transformation passes for each of the 16 blocks resulting from splitting each 64-bit block.
  - **AES** – (*Advanced Encryption Standard*) uses a fixed-size block of 128 bits with a key size of 128, 192, or 256 bits. The National Institute of Standards and Technology (NIST) approved the AES algorithm in December 2001, and the U.S. government uses AES to protect classified information.
  - **Blowfish**
- **Asymmetric Encryption** – Asymmetric encryption algorithms use two keys to encrypt data. One key is public, and the other is private. In the public key encryption system, anyone can encrypt the message using the recipient's public key, and only the recipient can decrypt the message using their private key. Parties exchange secure messages without needing a pre-shared key. Asymmetric algorithms are more complex, resource-consuming, and slower to execute because they are based on complex mathematical operations.
  Examples:
  - **RSA** (*Rivest–Shamir–Adleman*) – uses the product of two very large prime numbers with a length between 100 and 200 digits. Web browsers use RSA to establish a secure connection.

- ***Diffie-Hellman*** – provides a method of electronic key exchange. Secure protocols such as SSL (Secure Sockets Layer), TLS (Transport Layer Security), SSH (Secure Shell), and IPsec (Internet Protocol Security) use Diffie-Hellman..
- ***ElGamal*** – uses the U.S. government standard for digital signatures.
- **ECC** (*Elliptic Curve Cryptography*) – uses elliptic curves as part of the algorithm. In the United States, the NSA uses ECC for digital signature generation and key exchange.

VII. Best Practices and Cybersecurity Guidelines

People are the first line of defense in cybersecurity, and an institution is only as strong as its weakest link. Investments in technology will not make a significant difference in the fight against cybercriminals if the people in the organization are not trained. Every employee in an institution must be aware of security policies and implement them in day-to-day activities.

Security awareness should be an ongoing process because new threats and technologies continually emerge. Building an effective cybersecurity culture requires continuous effort and the involvement of all members of the institution.

7.1. Security policies

A security policy establishes security objectives, rules of behavior, and system requirements. Security policies inform users, employees, and managers about the institution's requirements for protecting technological and information assets. A security policy specifies the mechanisms needed to meet security requirements.

The following policies can be provided as examples: identification and authentication policies, policies for creating, using, and changing passwords, acceptable use policies, remote access policies, maintenance policies, incident management policies, etc.

An extensive list of general or specific policies regarding information security, applications, servers, or networks, as well as a series of templates that can be used to create such policies, can be found at SANS[2].

7.2. Security standards

Standards help employees maintain consistency in the use of institutional resources, improve efficiency, and streamline the maintenance and troubleshooting of information resources. One of the most important security principles is consistency. For this reason, it is necessary for organizations to establish standards. Each organization develops standards that support its unique operating environment.

ISO/IEC 27000 represents a series of information security standards designed to help organizations improve their security. Published by ISO and IEC, the ISO 27000 standards establish the requirements of a comprehensive ISMS (Information Security Management System). An ISMS consists of all administrative, technical, and operational measures that address information security in an organization.

[2] Information Security Policy Templates | SANS Institute (Septembrie, 2023)

ISO 27000 standard covers 12 independent domains that provide the foundation for developing security standards and effective security management practices in organizations. It helps facilitate communication between organizations in the following areas: risk assessment, security policy, organization of information security, asset management, human resource security, physical and environmental security, communication and operations management, acquisition, development, and maintenance of information systems, access control, security incident management, business continuity management, compliance with security policies, standards, and regulations.

7.3. Cybersecurity Security Guides and Best Practices

Guides are lists of suggestions on how things can be done more efficiently and securely. They are similar to standards but are much more flexible and are not usually mandatory. Guides define how standards are developed and ensure adherence to general security policies. There are numerous guides available, such as NIST Computer Resource Center, NSA Security Configuration Guides, ENISA Guidelines, etc.

Standards and some of the most useful guides allow the identification of best practices in cybersecurity. These may refer to:

- **Risk assessment**: Understanding the importance of assets to be protected and determining the amounts allocated for security.
- **Security policy creation**: Clearly stipulating the rules of the institution, responsibilities, and expectations.
- **Implementation of physical security measures**: Restricting access to areas where important resources are located.
- **Implementation of human resource security measures**: Properly screening employees.
- **Regular backup creation and testing**: Ensuring data recovery solutions are effective.
- **Security updates implementation**: For operating systems, services, and applications.
- **Access control:** Configuring roles and privilege levels for users.
- **Implementation of strong authentication services**: Enhancing user verification.
- **Periodic testing of incident response**: Preparing for emergency scenarios.
- **Implementation of network monitoring and management tools**: Ensuring network security.
- **Implementation of network security equipment**: Such as routers and firewalls.
- **Implementation of security solutions for computing systems**: Such as anti-malware and antivirus programs.
- **User and employee training**: Raising awareness about cybersecurity.
- **Encryption of important data**: Ensuring data confidentiality.

7.4. Security procedures

Procedures are longer and more detailed documents than standards and guides. They include implementation details that typically provide step-by-step instructions and graphical representations to illustrate how to perform a specific activity.

Any organization or institution should use a set of procedures to maintain consistency in conducting activities in a secure environment. All employees should be educated and informed about security procedures and the actions that need to be taken in the event of a security breach.

VIII. Incident Response and Disaster Recovery

Even after implementing all necessary security measures, it is almost inevitable that a security breach will occur at some point. The existence of a rapid and efficient response plan can make the difference between an incident with minor consequences and a major disaster. With a well-organized approach, damage can be limited (minimizing the impact), and the time and costs of recovery can be reduced.

8.1. Incident Response

Incident response is a set of procedures that an organization must follow in the event of a security incident or the emergence of cyber threats. This response includes several stages:

8.1.1. Preparation

An organization needs to develop an incident response plan and establish a CSIRT (Computer Security Incident Response Team) to manage the response. Depending on the size of the organization and the complexity of threats, the roles of this team include incident detection, incident analysis, incident response, communication and coordination with other teams, documentation and reporting, prevention of future incidents, employee training and awareness, and continuous monitoring of the IT environment to enhance security.

8.1.2. Detection and analysis

Proper detection involves identifying when an incident occurred, what data and systems were involved. Detection begins when someone discovers an incident.

An organization may have the most sophisticated detection systems, but they are worthless if administrators do not monitor logs and alerts. Once a security incident is detected, notifications are sent to management and those responsible for data and systems so they can be involved in remediation and repair.

After detection, it is necessary to confirm whether there is indeed a cybersecurity incident. This may involve further investigations to determine the nature and extent of the incident. Incident analysis helps identify the source, scope, impact, and details of the incident, including the data breach. The organization may decide to bring in an expert team to conduct a forensic investigation. A detailed investigation can be essential for a complete understanding of the incident.

8.1.3. Isolation, Eradication, and Recovery

An essential part of responding to a cybersecurity incident is isolating it to prevent its spread to other systems or resources within the organization and to reduce the impact and damage caused by it. For example, disconnecting the affected system from the network can stop the leakage of information.

After identifying and isolating the security breach, the next stage is the incident response phase, where measures are taken to eradicate (eliminate) the effects of the incident (e.g., removing malware).

This stage is followed by recovery, which involves remediation of affected resources and restoring systems and data to their original state.

8.1.4. Incident Follow-Up

After successfully managing the incident and restoring systems to their normal state, the final step of the process is responsible for addressing all aspects related to the incident. This includes identifying the cause and impact of the incident, the measures taken, and analyzing the lessons learned. It involves determining actions to prevent the recurrence of the incident or the occurrence of similar incidents, improving preventive measures, and minimizing the impact of such incidents.

All these aspects are documented and reported to enhance the organization's security measures.

8.2. Disaster Recovery

A disaster is any human or natural event that causes damage to goods or properties and affects an organization's ability to continue operations. There are two types of disasters: natural and human-induced.

Natural disasters vary by location and are difficult to predict:

- geological disasters (e.g., earthquakes, landslides, volcanic eruptions, and tsunamis)
- Meteorological disasters (e.g., hurricanes, tornadoes, blizzards, lightning, and hail)
- Health disasters (e.g., quarantine and pandemics)
- Other disasters (e.g., fires, floods, solar flares, and avalanches)

Human-induced disasters involve people or organizations and fall into the following categories:

- Workplace events (e.g., strikes)
- Socio-political events (e.g., vandalism, blockades, protests, sabotage, terrorism, and war)
- Material events (e.g., substance spills or fires)
- Utility disruptions (electricity supply), communication interruptions, fuel, and radioactive disasters.

When a disaster occurs, to ensure that critical systems are operational, an organization must activate a Disaster Recovery Plan (DRP). This plan includes all activities needed to assess, save, repair, and restore affected facilities or assets. Additionally, a DRP must identify critical processes and systems to prioritize their restoration during the recovery process.

Disaster recovery measures aim to minimize the effects of a disaster so that operations can be resumed. There are three types of disaster recovery measures:

- Preventive measures – aim to identify and reduce risks
- Detection measures – discover unwanted events and potential threats
- Corrective measures – restore the system after a disaster or event

Despite the best efforts to prevent disasters and data loss, it is impossible to account for every scenario. Hence, having a Business Continuity Plan (BCP) is critical, regardless of the circumstances.

A business continuity plan is more comprehensive than a DRP as it may involve relocating critical systems to another location while repairing the original headquarters. In such a scenario, employees will continue to perform all business processes in an alternative manner until normal operations are resumed.

IX. Bibliography

[1] **Cisco Skills For All**

Introduction to Cybersecurity

https://skillsforall.com/course/introduction-to-cybersecurity?userLang=en-US

Cybersecurity Essentials

https://skillsforall.com/course/cybersecurity-essentials?userLang=en-US

[2] **Coursera**

Palo Alto Networks Cybersecurity Foundation

https://www.coursera.org/learn/palo-alto-networks-cybersecurity-foundation-a

Activity 1

Exploring Social Engineering Techniques

Context

The cyber space in general, and the Internet in particular, presents a series of characteristics that make them hostile to unsuspecting users, exposing them to various threats. Attackers will attempt to compromise systems, networks, services, applications, and user data by using various deception and social engineering techniques.

Risky online behavior will facilitate the actions of attackers and pose a threat to one's own cybersecurity and the organization where users are employed, allowing unauthorized access to data, information systems, and network infrastructure. Personally Identifiable Information (PII), Protected Health Information (PHI), information protected by intellectual property rights (IP), and others can be exposed, destroyed, disclosed, or altered, causing personal, financial, or reputational harm.

Objectives

This practical activity is divided into two sections and aims, on the one hand, to identify online actions that could compromise security and privacy, and on the other hand, to explore social engineering techniques that manipulate and exploit people to make them perform certain actions or disclose specific information:

Part 1. Discovering the characteristics of one's own online behavior

Part 2. Exploring social engineering techniques

Part 1. Discovering the characteristics of one's own online behavior

1.1 Answer the following questions honestly and note the points you obtain for each question.

What kind of information do you share on social media? ____________

(3 points) Everything; I rely on social media to stay connected with friends and family.

(2 points) Articles and news that I find or read.

(1 point) It depends; I filter what information I share and with whom.

(0 points) Nothing; I don't use social media..

When creating a new account for an online service: ____________

(3 points) Reuse the same password used for other services to remember it more easily.

(3 points) Create a password as easy as possible, so you can remember it.

(1 point) Create a very complex password and store it using a password management service.

(1 point) Create a new password, similar but different from a password used for another service.

(0 points) Create a strong new password.

When receiving an email with links to other websites: ____________

(0 points) Do not click the link because you never click on links sent via email.

(3 points) Click the links because the email server has already scanned the email.

(2 points) Click all links if the email is from a person you know.

(1 point) Hover over links to check the destination URL before accessing them.

A pop-up window appears while visiting a website. According to the displayed text, your computer is in danger, and you should download and install a diagnostic program to be safe: __________

(3 points) Download and install the program to keep your computer safe.

(3 points) Check the pop-up windows and hover over the link to verify its validity.

(0 points) Ignore the message, ensuring you don't click on it or download the program, and close the site.

When you need to log in to your bank's website to perform a specific action: __________

(3 points) Immediately enter your authentication details.

(0 points) Check the URL to ensure it is the institution you are looking for before entering any information.

(0 points) Do not use online banking services or any kind of online financial services.

Read about a program and decide to give it a try. You search the internet and find a trial version on an unknown site: ____________

(3 points) Quickly download and install the program.

(1 point) Look for more information about the program's creator before downloading it.

(0 points) Do not download or install the program.

On the way to work, you find a USB: ____________

(3 points) Take it and connect it to the computer to see what it contains.

(3 points) Take it and connect it to the computer to completely erase the content before reusing it.

(3 points) Take it and connect it to the computer to run an antivirus scan before using it for your own files.

(0 points) Do not take it.

You need to connect to the Internet and find an open Wi-Fi hotspot. You.: ____________

(3 points) Connect to it and use the internet.

(0 points) Do not connect to it and wait until you have a trusted connection.

(0 points) Connect to it and establish a VPN connection to a trusted server before sending information..

1.2. Add up the points obtained for each question and calculate the total score, then analyze your online behavior.
Observation: The higher the score, the less secure your online behavior.

Interpretation of results:

0: You are safe online.

0 – 3: You are relatively safe online, but you should change your behavior to be completely secure.

3 – 17: You have unsafe online behavior, and there is a high risk of compromise.

1.3. Consult the recommendations presented at the address *Discover Your Own Risky Online Behavior.pdf* (netacad.com) [1] and look for other online recommendations.

1.4. After analyzing your online behavior and identifying best practices, what changes would you make to protect yourself?

Part 2. Exploring Social Engineering Techniques

Social engineering is the term used to describe a broad collection of malicious activities through which attackers can obtain information about the information systems and network of an organization. Although it relies on deceiving users, it is one of the most effective attack methods because it exploits the fact that, in general, people tend to trust others and let their guard down. Regardless of how good the security measures are, how well the equipment, servers, and network systems are configured, and how restrictive access control techniques are, deception techniques cannot be prevented without proper awareness and training of users.

The National Center for Systems Security and Information Assurance (CSSIA) in the United States hosts an online interactive presentation of social engineering techniques at https://www.cssia.org/social_engineering/ [2 If the address is no longer valid, search with a search engine for "CSSIA Social Engineering Interactive."

Using the information provided by CSSIA, study some of the most common social engineering techniques, trying to learn more about what they are, how they are executed, what damage they can cause, and whether you have been a victim of such attacks or if you are susceptible to them. Look online for examples of cases where such techniques have affected major names in the field of internationally or nationally renowned organizations.

2.1. Study the Baiting, Shoulder Surfing, and Pretexting Techniques

Baiting exploits curiosity by offering external memory devices (e.g., USB) infected with malware or relying on victims' greed. Attackers promise various free items (music, movies, games) in exchange for sensitive information such as usernames or passwords on certain websites.

Shoulder Surfing - attackers using this technique observe the victim's actions while entering PIN codes or passwords, either up close, taking advantage of crowds, or from a distance using various viewing devices such as binoculars or even high-performance camera-equipped mobile phones.

Pretexting involves creating a scenario to deceive the victim into disclosing sensitive information. Often coupled with the technique called Impersonating, where the attacker pretends to be someone else (a client, hierarchical superior, etc.).

2.2. Study the Phishing/Spear Phishing and Whaling Techniques

The attacks in this category attempt to obtain personal information, credit card details, usernames, or passwords using emails or phone calls that persuade victims to click on links, open files infected with malware attached to emails, or visit malicious websites.

Spear phishing is a targeted form of phishing, focusing on a specific user or a chosen institution after a reconnaissance stage where data about the target is gathered to enable the configuration and customization of the attack.

Whaling is a phishing attack targeting a high-level individual who has access to more confidential information.

2.3. Study Scareware and Ransomware Techniques

Scareware is a technique that attempts to deceive the user by making them believe that their system (computer) is infected with malware.

Ransomware is an attack in which malware encrypts the user's computer data and demands payment of a sum of money to provide the key for decrypting the data.

2.4. Create a cybersecurity awareness poster

Using PowerPoint or a similar application, create a poster that will present various social engineering techniques that can be used to gain unauthorized access to the systems, infrastructure, and data of an institution/organization, as well as how these attacks can be avoided.

Bibliography

[1] Discover Your Own Risky Online Behavior.pdf (netacad.com)
https://contenthub.netacad.com/legacy/I2CS/2.1/en/course/files/3.2.2.3%20Lab%20-%20Discover%20Your%20Own%20Risky%20Online%20Behavior.pdf

[2] Social Engineering Interactive - CSSIA: NSF ATE Center
https://www.cssia.org/social_engineering/

Practical Activity 2

Access Control Configuration

Context

Access control is the process of restricting access rights to control the use of networks, equipment, data, services, and applications. Administrative access control involves three essential security services: authentication, authorization, and accounting. Authentication and authorization are distinct security processes in identity and access management. Authentication is the process of confirming the identity of a user or system to prevent unauthorized access, using passwords or other identification methods. Authorization is the process of granting or denying access to specific resources.

The following network architecture is considered:

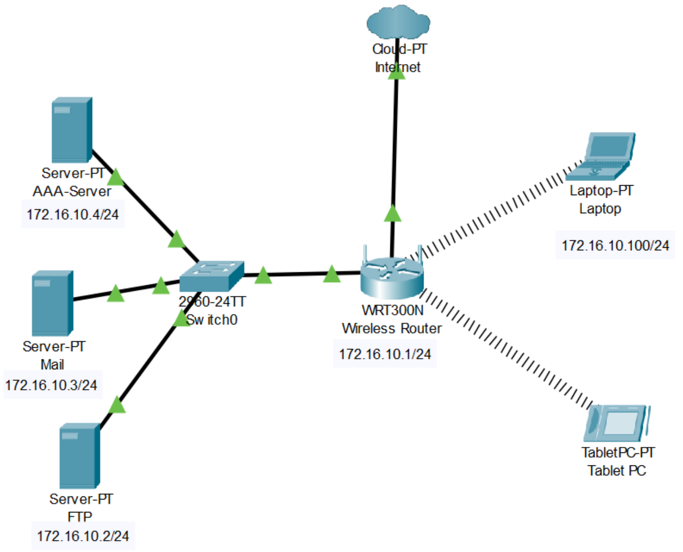


Fig. 1. Network architecture

Objectives

In this activity, the Cisco Packet Tracer network simulation software [1, 2] will be used to configure authentication and authorization for accessing network services such as wireless access, email, or file servers. For a thorough understanding of all aspects discussed in this activity, it is recommended to refer to [3-5].

The activity includes the following parts:

Part 1. Configuration and use of AAA services

Part 2. Configuration and use of email services

Part 3. Configuration and use of FTP (File Transfer Protocol) services

Before configuring access control, authentication, and authorization, the basic IP configuration of the equipment (IP addresses, subnet masks, default gateway routers, etc.) must be performed according to the information provided in the diagram in Figure 1.

Part 1. Configuration and use of AAA services

1.1 Configuration of the Radius service and user accounts on the AAA server

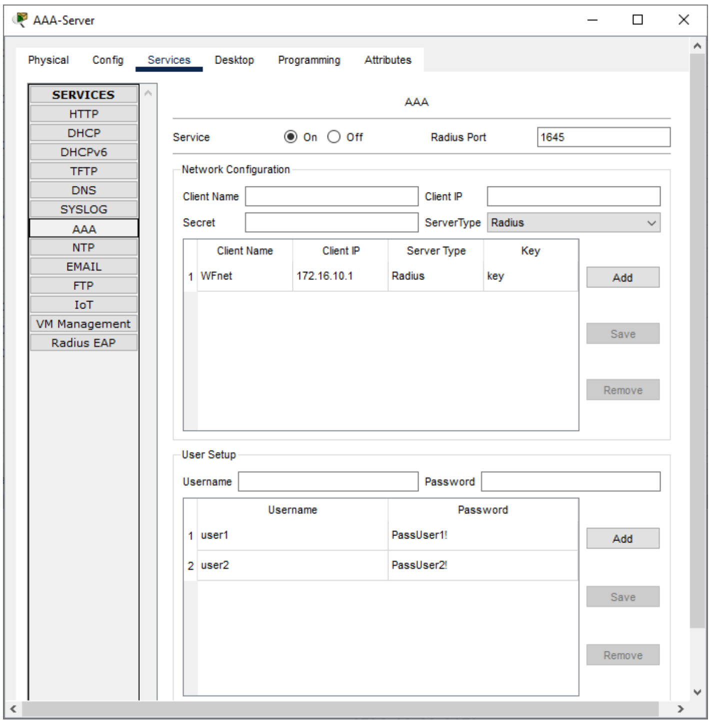
a. Open the AAA Server configuration window (left-click on the server icon) and select AAA from the Services panel on the left.

b. Start the service by selecting the On radio button.

c. In the Network Configuration section, fill in the client name (e.g., WFnet), the client's IP address (e.g., the wireless router address, 172.16.10.1), and the secret code (e.g., key).

d. In the User Setup section, add the following usernames and passwords:

- user1/PassUser1!
- user2/PassUser2!



AAA

Service ● On ○ Off Radius Port 1645

Network Configuration

Client Name Client IP

Secret ServerType Radius

	Client Name	Client IP	Server Type	Key
1	WFnet	172.16.10.1	Radius	key

User Setup

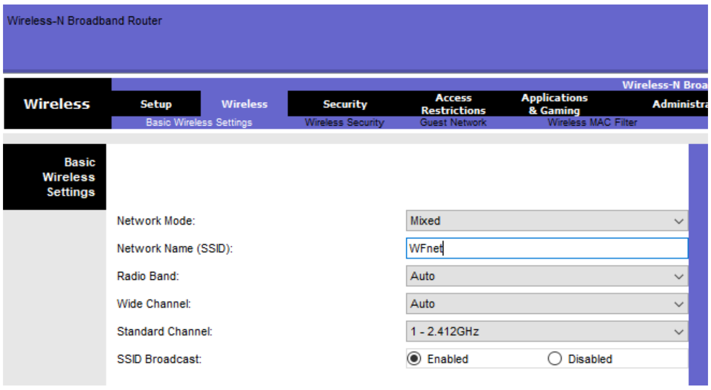
Username Password

	Username	Password
1	user1	PassUser1!
2	user2	PassUser2!

Fig. 2. Configuration of AAA services on server

1.2 Configuration of the wireless router

a. Open the configuration window of the Wireless Router (left-click on the router icon) and select Basic Wireless Settings from the Wireless panel. Fill in the network name - SSID (e.g., WFnet).



Network Mode: Mixed

Network Name (SSID): WFnet

Radio Band: Auto

Wide Channel: Auto

Standard Channel: 1 - 2.412GHz

SSID Broadcast: ● Enabled ○ Disabled

Fig. 3. SSID configuration

b. Save the configuration changes by pressing the Save Settings button at the bottom of the page.

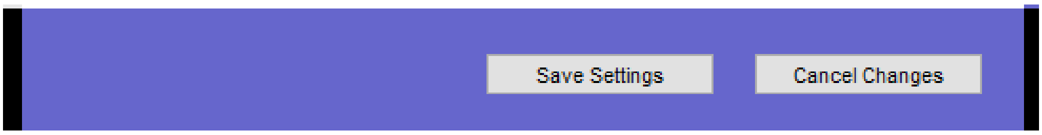
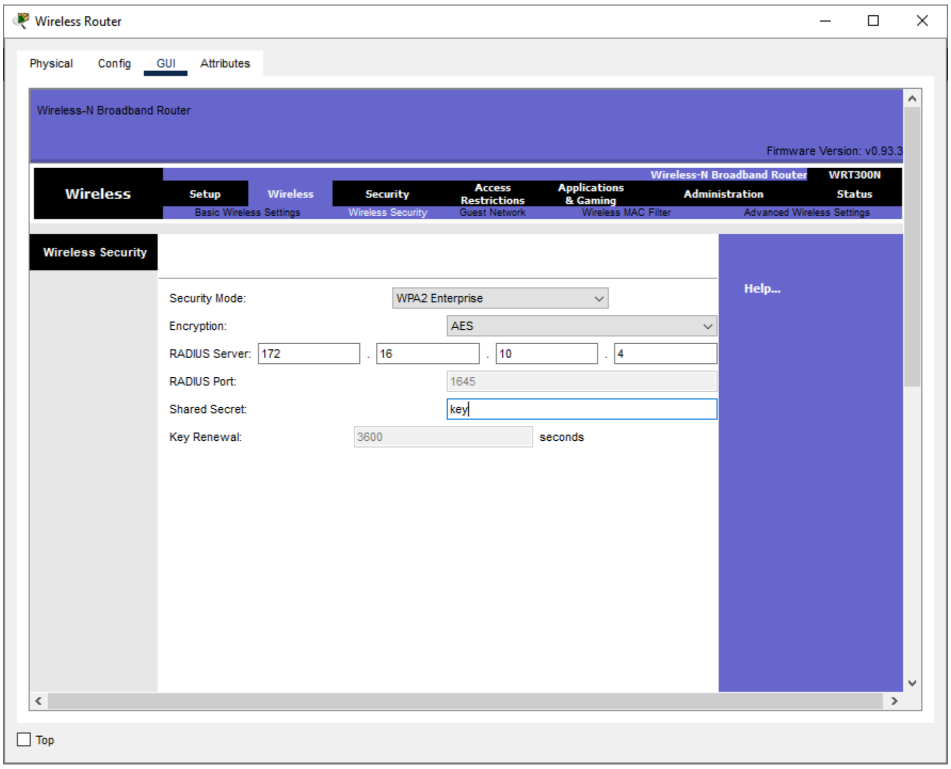


Fig.4. Saving the *wireless router configuration*

c. Select the WPA2 Enterprise security mode in the Wireless Security panel under the Wireless section. Fill in the necessary information for connecting to the AAA Radius server (e.g., server 172.16.10.4 and the previously configured secret code on the server, key).



Security Mode: WPA2 Enterprise

Encryption: AES

RADIUS Server: 172 . 16 . 10 . 4

RADIUS Port: 1645

Shared Secret: key

Key Renewal: 3600 seconds

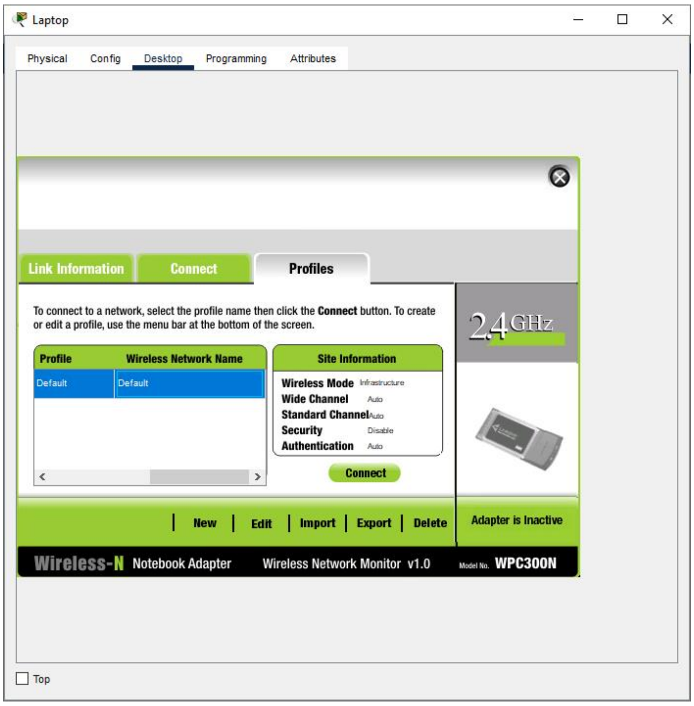
Fig. 5. Configuration of WPA2 Enterprise Wireless Security

d. Save the configuration changes by pressing the Save Settings button at the bottom of the page.

1.3 Configuration of wireless authentication on the laptop and tablet

a. Open the configuration window of the laptop (left-click on the laptop icon) and select the PC Wireless application from the Desktop panel.

b. In the dialog window, select the Profiles panel, choose the existing profile (Default), and press Edit.



To connect to a network, select the profile name then click the **Connect** button. To create or edit a profile, use the menu bar at the bottom of the screen.

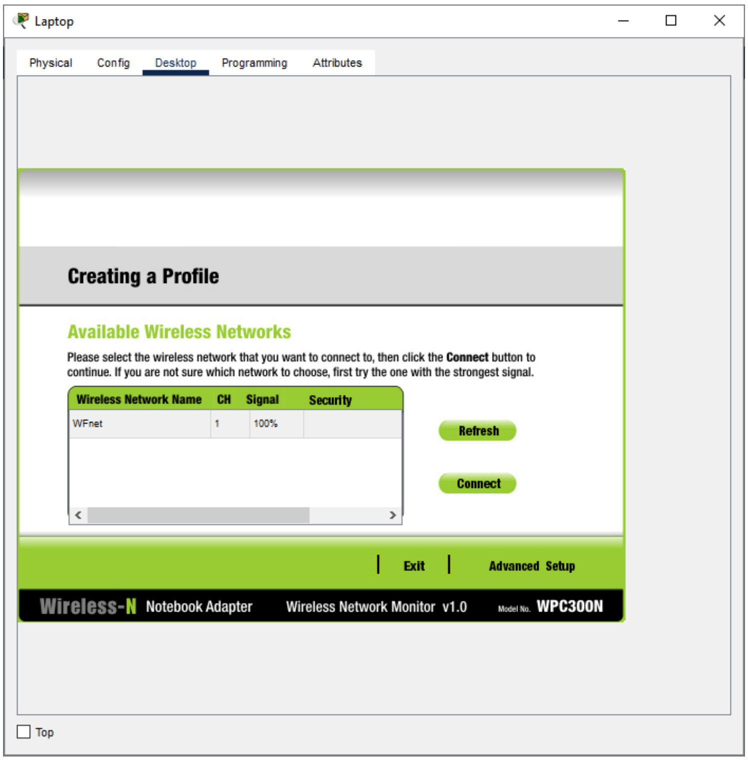
Profile	Wireless Network Name
Default	Default

Site Information

Wireless Mode	Infrastructure
Wide Channel	Auto
Standard Channel	Auto
Security	Disable
Authentication	Auto

Fig. 6. Editing the wireless profile

c. From the list of available wireless networks, select the desired network (e.g., WFnet) and press the Advanced Setup button.



Creating a Profile

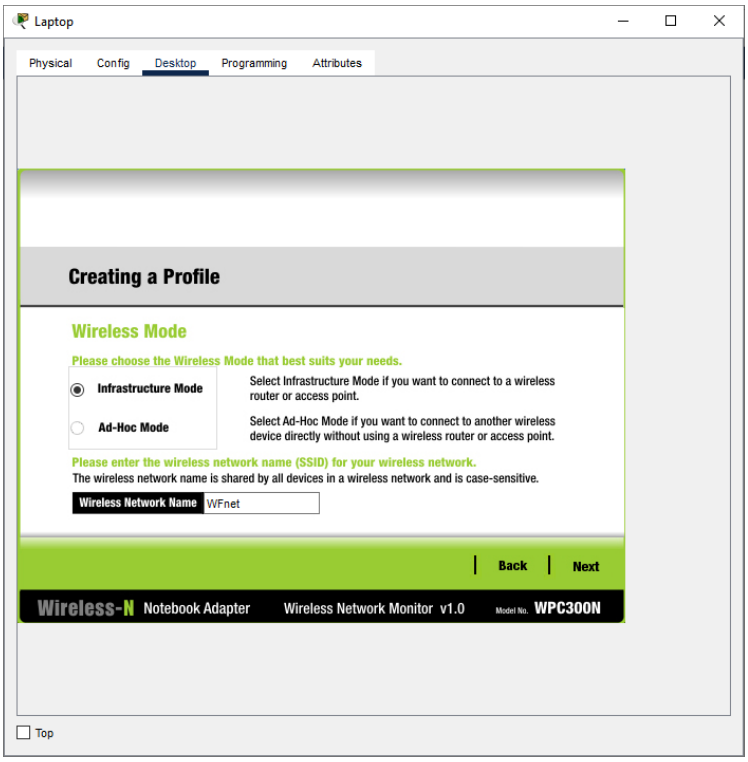
Available Wireless Networks

Please select the wireless network that you want to connect to, then click the **Connect** button to continue. If you are not sure which network to choose, first try the one with the strongest signal.

Wireless Network Name	CH	Signal	Security
WFnet	1	100%	

Fig. 7. Advanced Configuration of the Wireless Profile

d. Choose Infrastructure Mode, enter the wireless network name (if not already filled in - e.g., WFnet), and press Next.



Creating a Profile

Wireless Mode

Please choose the Wireless Mode that best suits your needs.

● Infrastructure Mode — Select Infrastructure Mode if you want to connect to a wireless router or access point.

○ Ad-Hoc Mode — Select Ad-Hoc Mode if you want to connect to another wireless device directly without using a wireless router or access point.

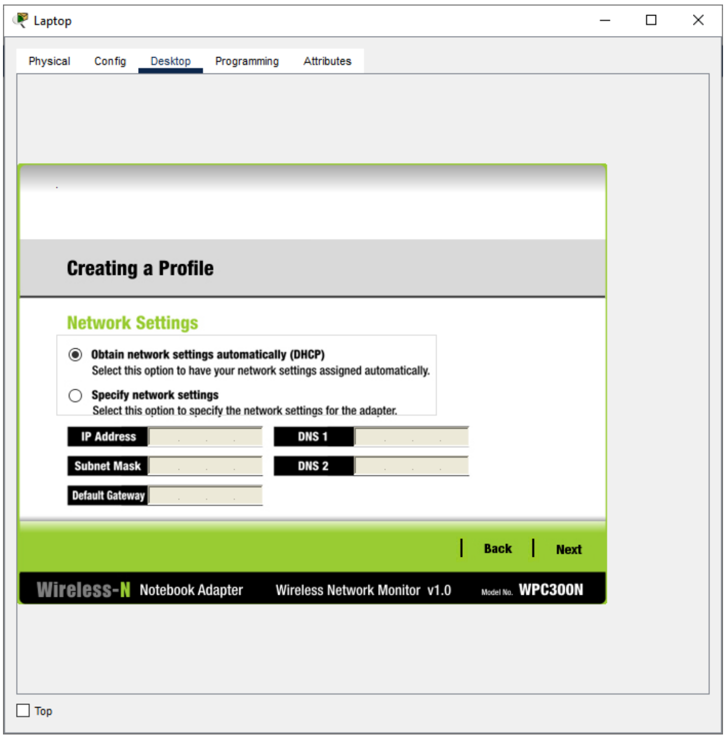
Please enter the wireless network name (SSID) for your wireless network.

The wireless network name is shared by all devices in a wireless network and is case-sensitive.

Wireless Network Name: WFnet

Fig. 8. Selecting the Wireless Mode

e. Configure network settings by, for example, selecting automatic IP configuration via DHCP, and press Next.



Creating a Profile

Network Settings

● Obtain network settings automatically (DHCP)
Select this option to have your network settings assigned automatically.

○ Specify network settings
Select this option to specify the network settings for the adapter.

IP Address

Subnet Mask

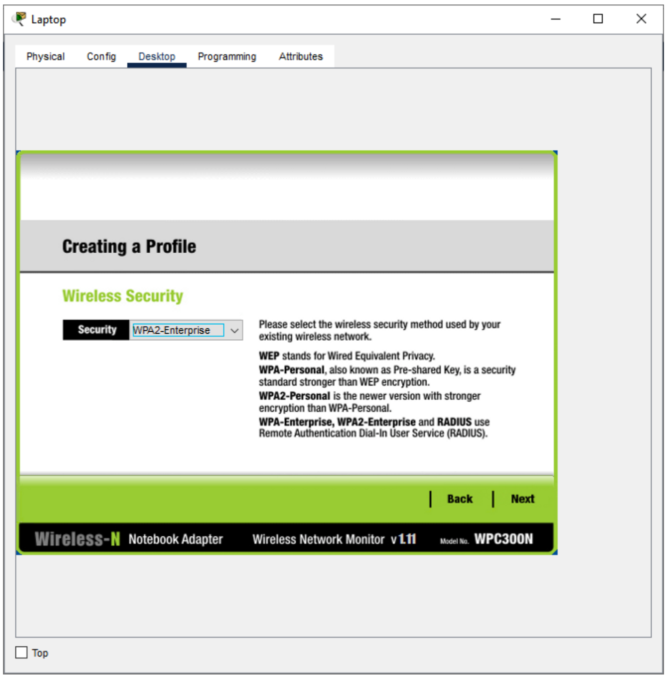
Default Gateway

DNS 1

DNS 2

Fig. 9. Selecting the IP Configuration Model

f. Select the WPA2-Enterprise security protocol and press Next.



Creating a Profile

Wireless Security

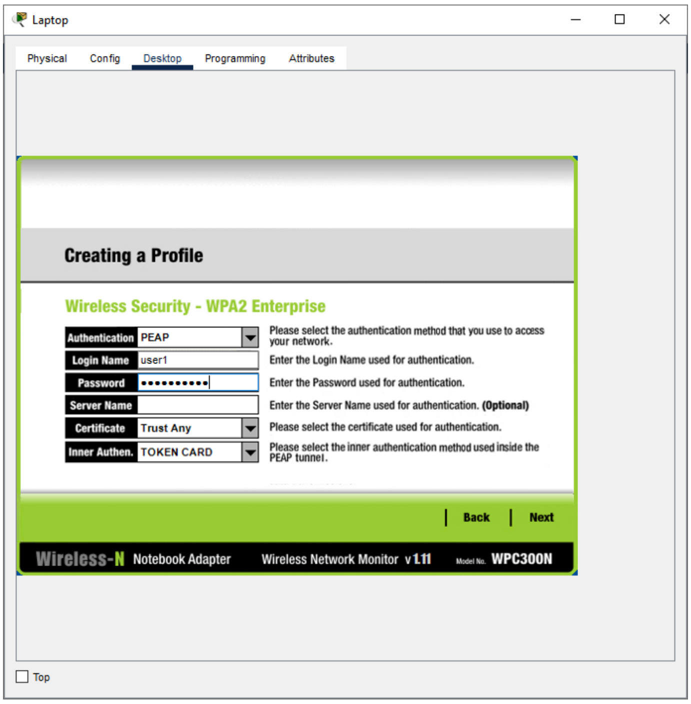
Security: WPA2-Enterprise

Please select the wireless security method used by your existing wireless network.

WEP stands for Wired Equivalent Privacy.
WPA-Personal, also known as Pre-shared Key, is a security standard stronger than WEP encryption.
WPA2-Personal is the newer version with stronger encryption than WPA-Personal.
WPA-Enterprise, WPA2-Enterprise and **RADIUS** use Remote Authentication Dial-In User Service (RADIUS).

Fig. 10. Selecting the Security Protocol

g. Fill in the security protocol configuration details, including the username (e.g., user1) and password (e.g., PassUser1!), and press Next.



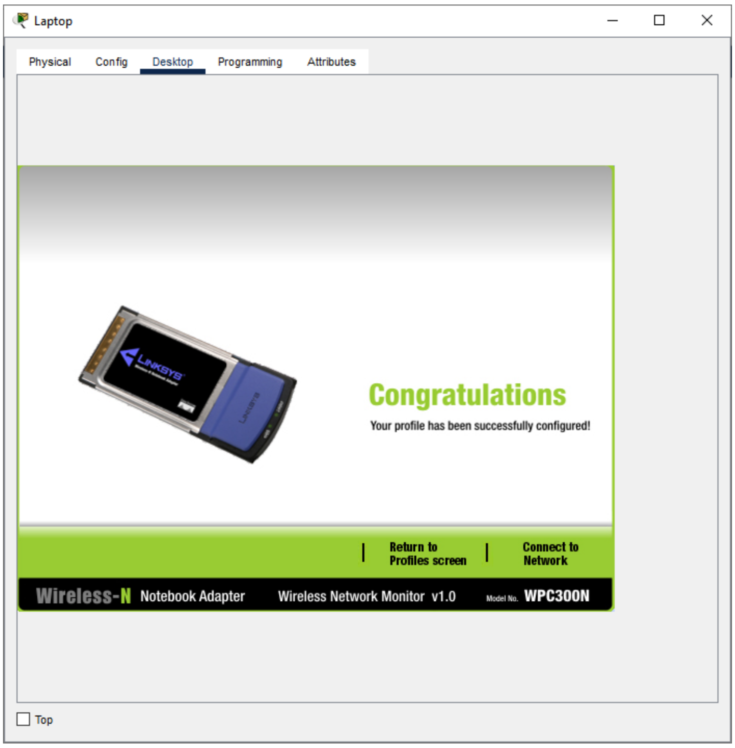
Creating a Profile

Wireless Security - WPA2 Enterprise

Authentication	PEAP	Please select the authentication method that you use to access your network.
Login Name	user1	Enter the Login Name used for authentication.
Password	••••••••••	Enter the Password used for authentication.
Server Name		Enter the Server Name used for authentication. **(Optional)**
Certificate	Trust Any	Please select the certificate used for authentication.
Inner Authen.	TOKEN CARD	Please select the inner authentication method used inside the PEAP tunnel.

Fig. 11. Configuring the Security Protocol

h. Confirm and save the configuration, then press Connect to Network.

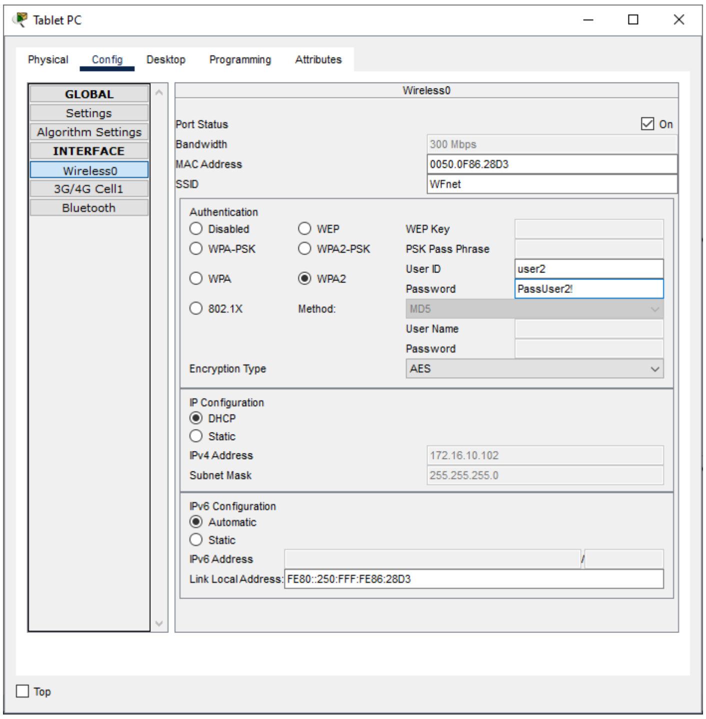


Congratulations

Your profile has been successfully configured!

Fig. 12. Connecting to the Network

i. Perform a similar configuration for the tablet, filling in the necessary information (see the configuration from previous steps) in the Config panel.



Wireless0

Port Status ☑ On

Bandwidth 300 Mbps

MAC Address 0050.0F86.28D3

SSID WFnet

Authentication

○ Disabled ○ WEP WEP Key

○ WPA-PSK ○ WPA2-PSK PSK Pass Phrase

User ID user2

○ WPA ● WPA2 Password PassUser2!

○ 802.1X Method: MD5

User Name

Password

Encryption Type AES

IP Configuration

● DHCP

○ Static

IPv4 Address 172.16.10.102

Subnet Mask 255.255.255.0

IPv6 Configuration

● Automatic

○ Static

IPv6 Address /

Link Local Address: FE80::250:FFF:FE86:28D3

Fig. 13. Configuring and Connecting the Tablet to the Wireless Network

j. Verify if the devices have connected to the wireless network and have received IP addresses from the 172.16.10.0/24 segment.

Part 2. Configuration and use of email services

2.1 Enabling email services and configuring user email accounts

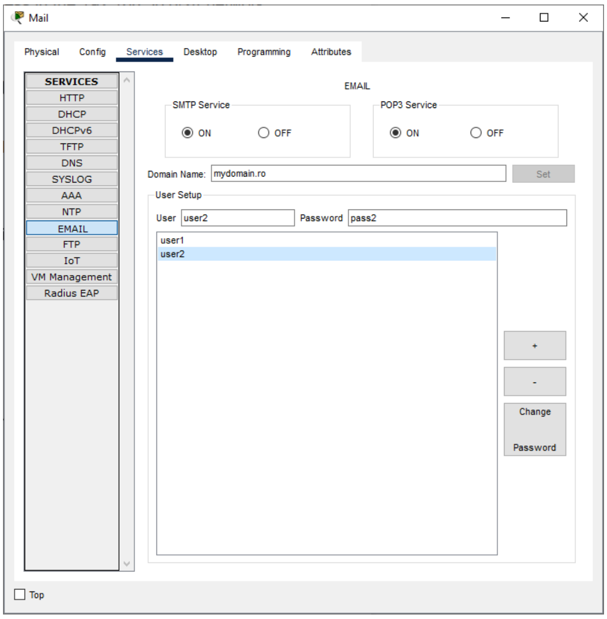
a. Open the Mail Server configuration window (left-click on the server icon) and select EMAIL from the Services panel on the left.

b. Start the SMTP and POP3 services.

c. Set the domain to mydomain.ro.

d. Create the following accounts:

- user1/pass1
- user2/pass2



EMAIL

SMTP Service: ● ON ○ OFF

POP3 Service: ● ON ○ OFF

Domain Name: mydomain.ro

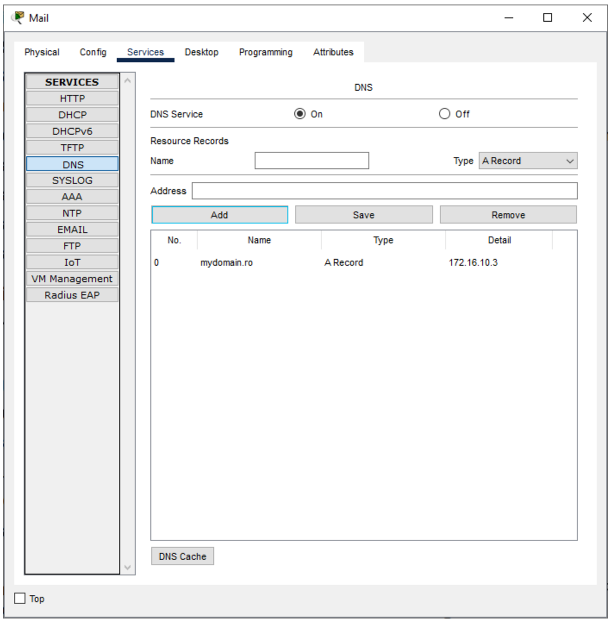
User Setup

User user2 Password pass2

user1
user2

Fig. 14. Email Services Configuration

Note: The domain mydomain.ro must already be registered on a DNS server, which can be the same as the email server:



DNS

DNS Service: ● On ○ Off

Resource Records

Name: Type: A Record

Address:

No.	Name	Type	Detail
0	mydomain.ro	A Record	172.16.10.3

Fig. 15. DNS Services Configuration

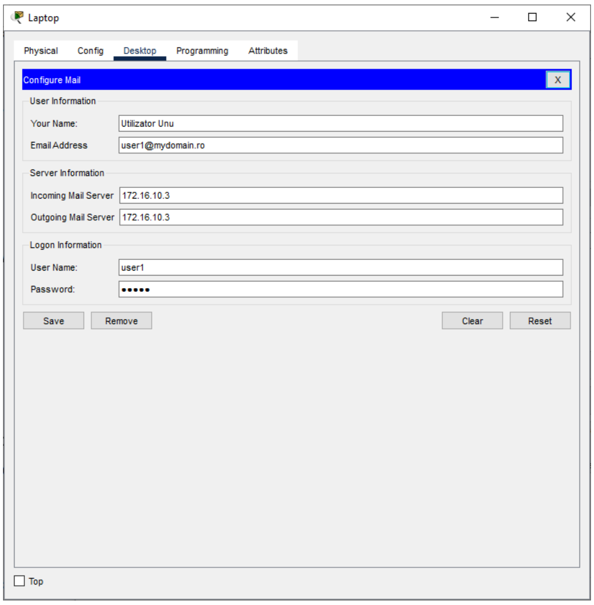
2.2 Configuring email clients on the laptop and tablet

a. Open the configuration window of the laptop (left-click on the laptop icon) and use the Email application from the Desktop panel.

b. Configure the email client with the following information and press the Save button:

- Username: Utilizator Unu
- Email address: user1@mydomain.ro
- Incoming mail server: 172.16.10.3
- Outgoing mail server: 172.16.10.3
- Username: user1

- Password: pass1



User Information

Your Name: Utilizator Unu

Email Address: user1@mydomain.ro

Server Information

Incoming Mail Server: 172.16.10.3

Outgoing Mail Server: 172.16.10.3

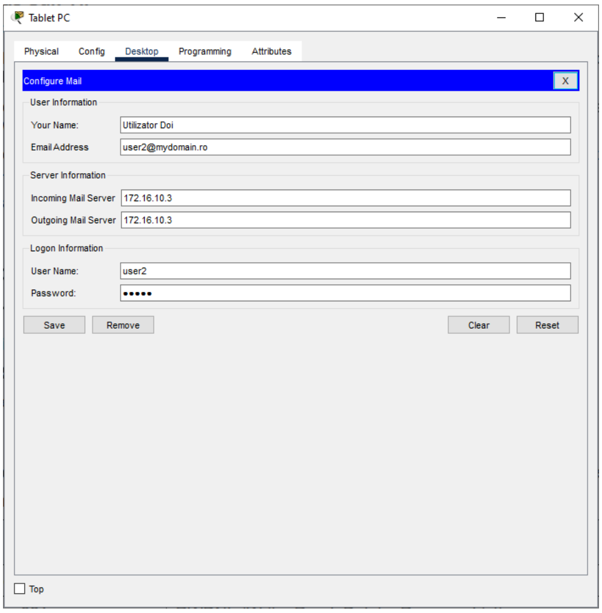
Logon Information

User Name: user1

Password: •••••

Fig. 16. Configuring the Email Client Application on the Laptop

c. Repeat the previous steps to configure the email client on the tablet for user2.



User Information

Your Name: Utilizator Doi

Email Address: user2@mydomain.ro

Server Information

Incoming Mail Server: 172.16.10.3

Outgoing Mail Server: 172.16.10.3

Logon Information

User Name: user2

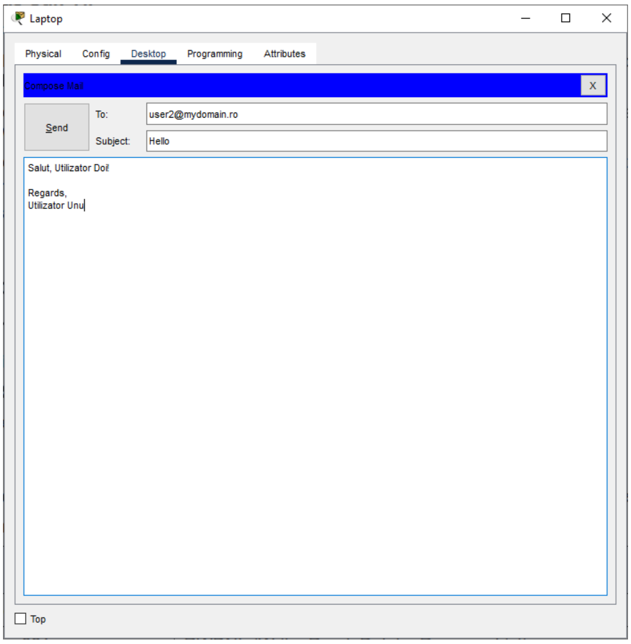
Password: •••••

Fig. 17. Configuring the Email Client Application on the Tablet

2.3 Sending and receiving an email message

a. Open the configuration window of the laptop (left-click on the laptop icon) and use the Email application from the Desktop panel.

b. Press the Compose button and compose an email message to user2@mydomain.ro, then press the Send button.



To: user2@mydomain.ro

Subject: Hello

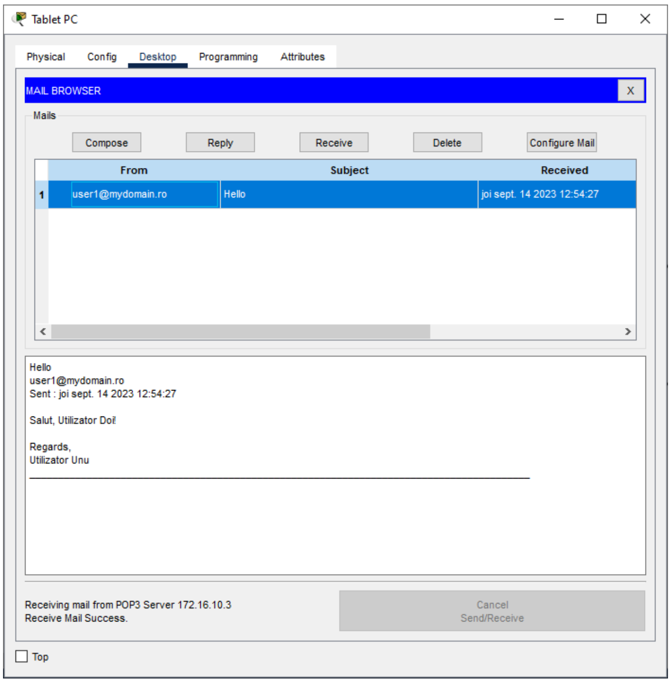
Salut, Utilizator Doi!

Regards,
Utilizator Unu

Fig. 18. Composing and Sending an Email Message

c. Open the configuration window of the tablet (left-click on the tablet icon) and use the Email application from the Desktop panel.

d. Press the Receive button.



	From	Subject	Received
1	user1@mydomain.ro	Hello	joi sept. 14 2023 12:54:27

Hello
user1@mydomain.ro
Sent : joi sept. 14 2023 12:54:27

Salut, Utilizator Doi!

Regards,
Utilizator Unu

Receiving mail from POP3 Server 172.16.10.3
Receive Mail Success.

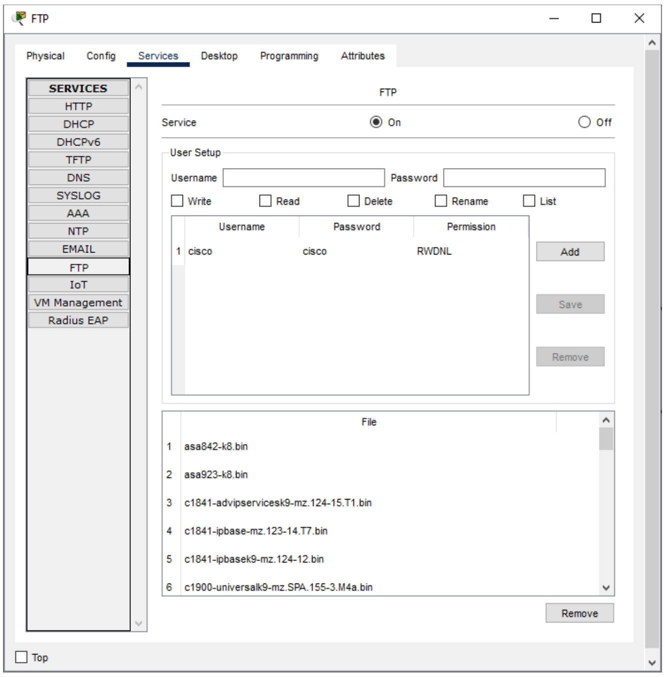
Fig. 19. Receiving an Email Message

Part 3. Configuration and use of file transfer services

3.1 Enabling FTP services

a. Open the File Server configuration window (left-click on the server icon) and select FTP from the Services panel on the left.

b. Start the FTP service.



FTP

Service ● On ○ Off

User Setup

Username: Password:

□ Write □ Read □ Delete □ Rename □ List

	Username	Password	Permission
1	cisco	cisco	RWDNL

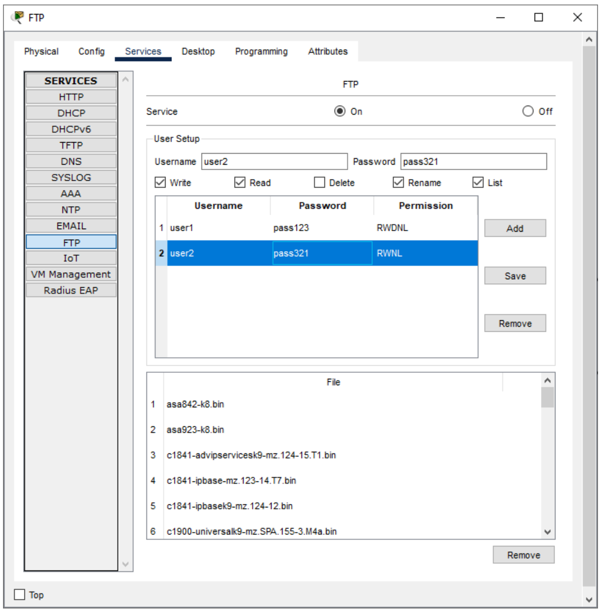
	File
1	asa842-k8.bin
2	asa923-k8.bin
3	c1841-advipservicesk9-mz.124-15.T1.bin
4	c1841-ipbase-mz.123-14.T7.bin
5	c1841-ipbasek9-mz.124-12.bin
6	c1900-universalk9-mz.SPA.155-3.M4a.bin

Fig. 20. Starting the File Transfer Service

3.2 Creating FTP user accounts

a. Create the following user accounts, establishing the privileges for each.

- user1/pass123/ RWDNL (*Write, Read, Delete, Rename, List*)
- user2/pass321/ RWNL (*Write, Read, Rename, List*)



FTP

Service ● On ○ Off

User Setup

Username: user2 Password: pass321

☑ Write ☑ Read □ Delete ☑ Rename ☑ List

	Username	Password	Permission
1	user1	pass123	RWDNL
2	user2	pass321	RWNL

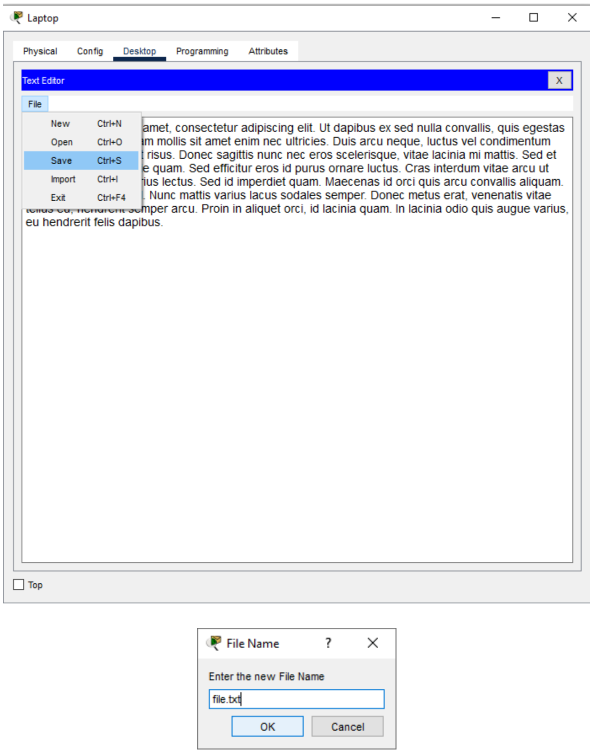
	File
1	asa842-k8.bin
2	asa923-k8.bin
3	c1841-advipservicesk9-mz.124-15.T1.bin
4	c1841-ipbase-mz.123-14.T7.bin
5	c1841-ipbasek9-mz.124-12.bin
6	c1900-universalk9-mz.SPA.155-3.M4a.bin

Fig. 21. Creating and Configuring User Account Privileges

3.3 Uploading and downloading files on the FTP server

a. Open the configuration window of the laptop (left-click on the server icon) and use the Text Editor application from the Desktop panel.

b. Compose a text file and save it with the name file.txt**.**



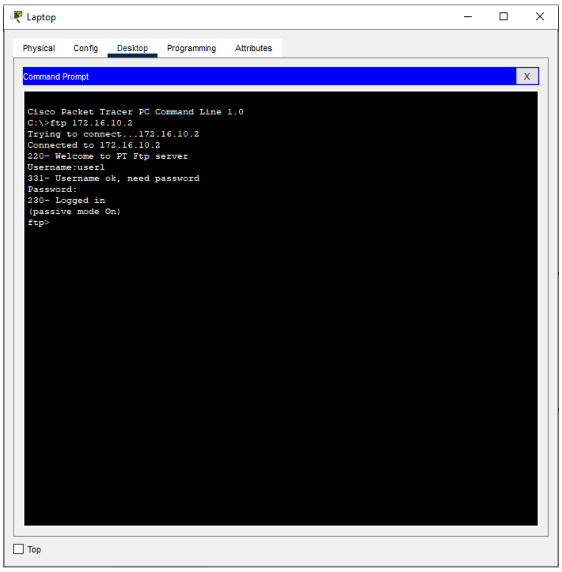
```
amet, consectetur adipiscing elit. Ut dapibus ex sed nulla convallis, quis egestas
m mollis sit amet enim nec ultricies. Duis arcu neque, luctus vel condimentum
risus. Donec sagittis nunc nec eros scelerisque, vitae lacinia mi mattis. Sed et
e quam. Sed efficitur eros id purus ornare luctus. Cras interdum vitae arcu ut
ius lectus. Sed id imperdiet quam. Maecenas id orci quis arcu convallis aliquam.
. Nunc mattis varius lacus sodales semper. Donec metus erat, venenatis vitae
mper arcu. Proin in aliquet orci, id lacinia quam. In lacinia odio quis augue varius,
eu hendrerit felis dapibus.
```

Enter the new File Name: file.txt

Fig. 22. Creating and Saving a Text File

c. Close the file editor, and open the command-line interface, Command Prompt.

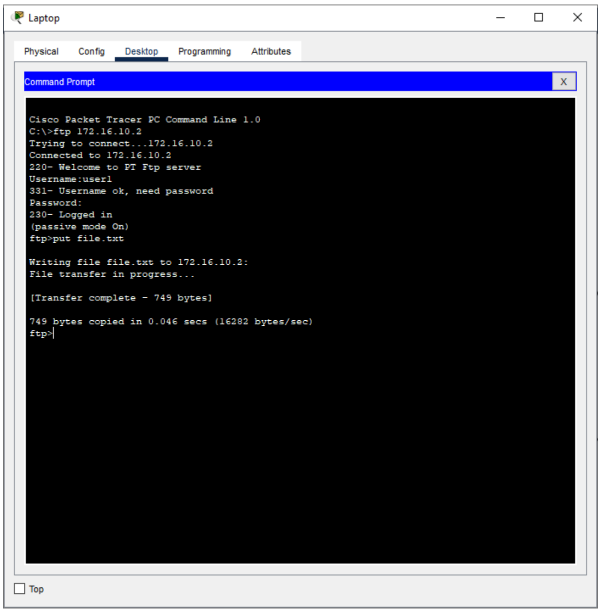
d. Enter the command ftp 172.16.10.2 and authenticate with the username user1 and password pass123.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ftp 172.16.10.2
Trying to connect...172.16.10.2
Connected to 172.16.10.2
220- Welcome to PT Ftp server
Username:user1
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Fig. 23. Connecting to the File Server

e. Use the put command to upload the file file.txt.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ftp 172.16.10.2
Trying to connect...172.16.10.2
Connected to 172.16.10.2
220- Welcome to PT Ftp server
Username:user1
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put file.txt

Writing file file.txt to 172.16.10.2:
File transfer in progress...

[Transfer complete - 749 bytes]

749 bytes copied in 0.046 secs (16282 bytes/sec)
ftp>
```

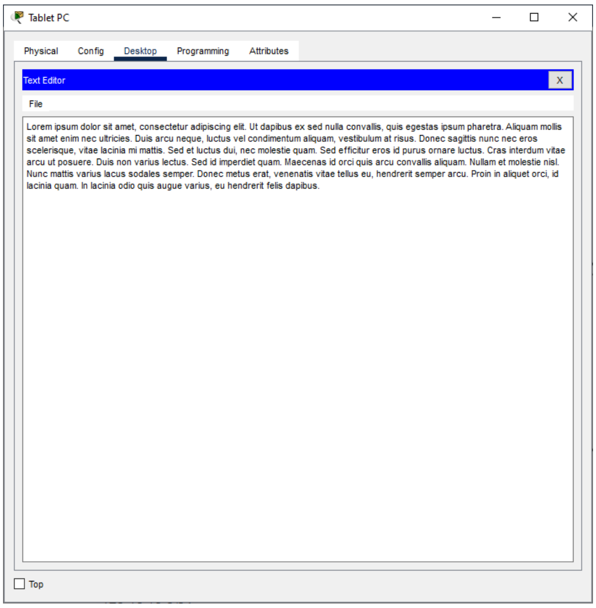
Fig. 24. Uploading a File to the File Server

f. Open the configuration window of the tablet (left-click on the tablet icon) and use the command-line interface, Command Prompt, from the Desktop panel.

g. Enter the command ftp 172.16.10.2 and authenticate with the username user2 and password pass321.

h. Use the get command to download the file file.txt.

i. Using the Text Editor text file editor, open the file file.txt and verify its content.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut dapibus ex sed nulla convallis, quis egestas ipsum pharetra. Aliquam mollis sit amet enim nec ultricies. Duis arcu neque, luctus vel condimentum aliquam, vestibulum at risus. Donec sagittis nunc nec eros scelerisque, vitae lacinia mi mattis. Sed et luctus dui, nec molestie quam. Sed efficitur eros id purus ornare luctus. Cras interdum vitae arcu ut posuere. Duis non varius lectus. Sed id imperdiet quam. Maecenas id orci quis arcu convallis aliquam. Nullam et molestie nisl. Nunc mattis varius lacus sodales semper. Donec metus erat, venenatis vitae tellus eu, hendrerit semper arcu. Proin in aliquet orci, id lacinia quam. In lacinia odio quis augue varius, eu hendrerit felis dapibus.

Fig. 25. Content of the File Downloaded from the File Server

3.4 Verifying user privileges

a. Open the configuration window of the tablet (left-click on the tablet icon) and use the command-line interface, Command Prompt, from the Desktop panel.

b. Enter the command ftp 172.16.10.2 and authenticate with the username user2 and password pass321.

c. Use the delete command to delete the file file.txt.

    Could this operation be performed? What is the reason?

d. Use the rename command to rename the file file.txt.

e. Repeat the file deletion operation (step c.) after connecting with the credentials of the user user1.

    Could this operation be performed? What is the reason?

Bibliography

[1] Cisco Packet Tracer - Networking Simulation Tool (netacad.com)
https://www.netacad.com/courses/packet-tracer

[2] Exploring Networking with Cisco Packet Tracer (skillsforall.com)
https://skillsforall.com/course/exploring-networking-cisco-packet-tracer?courseLang=en-US

[3] Packet Tracer: WPA2 Enterprise using RADIUS Server Configuration - YouTube
https://www.youtube.com/watch?v=L4VvqIBMdCU

[4] how to configure email server in packet tracer - YouTube
https://www.youtube.com/watch?v=D0N1EMQe9SA

[5] FTP Server Cisco Packet Tracer - YouTube
https://www.youtube.com/watch?v=MPTrbFzIn0Y

Practical Activity 3

Configuring the Security of a Wireless Router

Context

Security threats are numerous and can be classified into various categories, with attacks on infrastructure and network equipment being among the most common. One area with a high security risk is wireless networks, widely used today in both private and public environments. The reasons for their vulnerability are related to easy accessibility and the various weaknesses of network protocols and equipment, making them susceptible to unauthorized access, data interception, electronic interference, or Man-in-the-Middle attacks.

Best practices for the secure use of wireless networks include: changing default passwords for managing equipment, modifying the SSID identifier, creating and monitoring a guest network, using firewall equipment, filtering MAC addresses of clients, disabling remote management, etc. [1]

The following network architecture is considered:

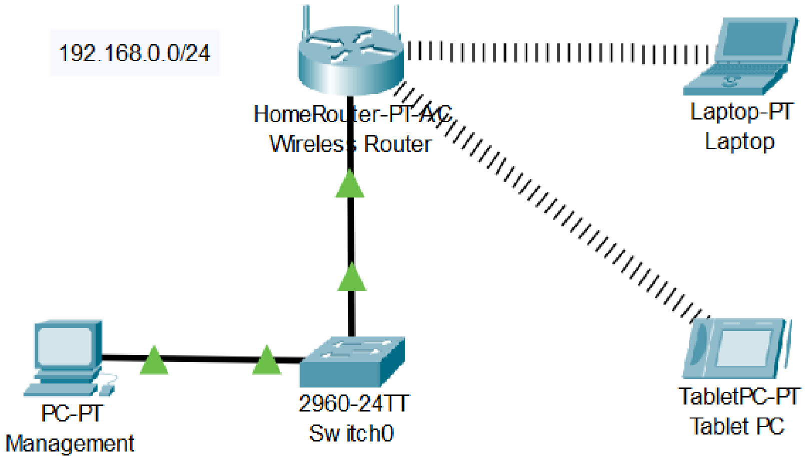


Fig. 1. Network architecture

Objectives

In this activity, security parameters on a wireless router will be configured. During the installation and setup of the router, existing security issues will be identified and addressed. Additionally, settings will be adjusted to strengthen the router and reduce the potential risks of attacks:

The activity consists of the following parts:

Part 1. Configuring basic settings for a wireless router

Part 2. Configuring the security of the router's network and wireless clients

Part 3. Checking wireless connectivity and security

Part 1. Configuring basic settings for a wireless router

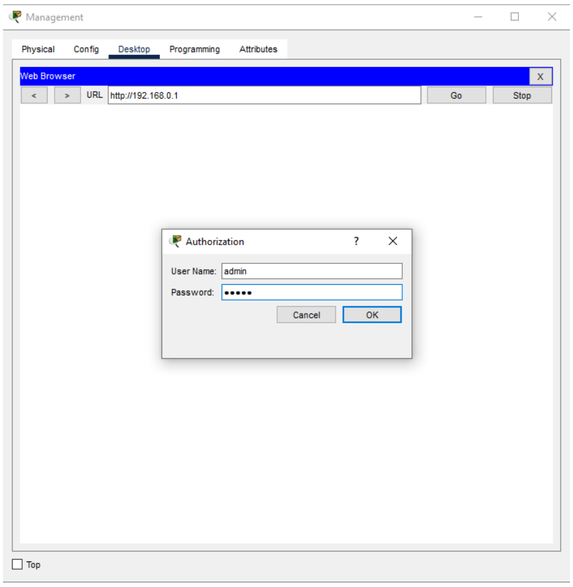
The initial configurations of network devices can pose a security risk. An attacker who manages to discover the IP address of a device may attempt to connect to it remotely. After finding default login information for administrative accounts on the Internet, they could try to take control of the device.

1.1 Changing the Default Router Administration Password

a. Open the management PC configuration window (left-click on the computer icon) and access the Web Browser application from the Desktop panel.

b. Connect the browser to the address 192.168.0.1 (the default LAN address of the wireless router).

c. Enter the default username and password (admin/admin) and click the OK button.



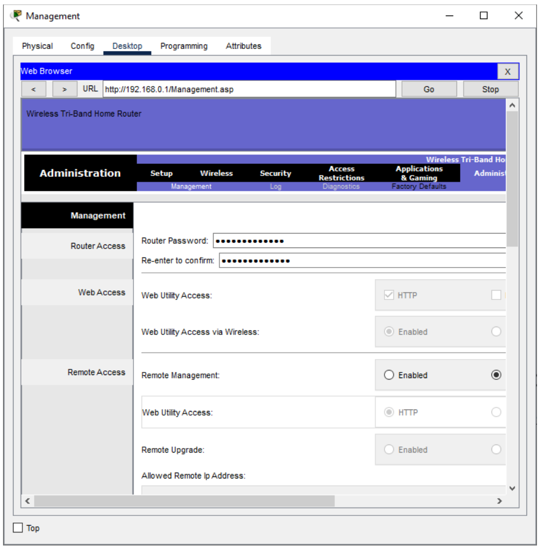
URL http://192.168.0.1

User Name: admin

Password: •••••

Fig. 2. Connecting to the Web Configuration Application of the Wireless Router

d. Navigate to the Administration section, enter and confirm a new password, following recommendations and best practices for creating a strong password [2] (e.g. **tH3NeWpass^wd**)



URL http://192.168.0.1/Management.asp

Router Password: ••••••••••••••

Re-enter to confirm: ••••••••••••••

Web Utility Access: ☑ HTTP □

Web Utility Access via Wireless: ● Enabled ○

Remote Management: ○ Enabled ●

Web Utility Access: ● HTTP ○

Remote Upgrade: ○ Enabled ○

Allowed Remote Ip Address:

Fig. 3. Changing the Administration Password of the Wireless Router

e. Save the configuration changes by clicking the "Save Settings" button at the bottom of the page.

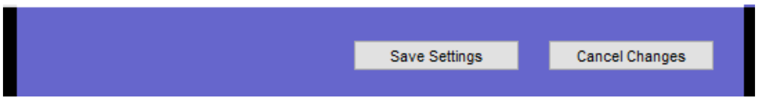
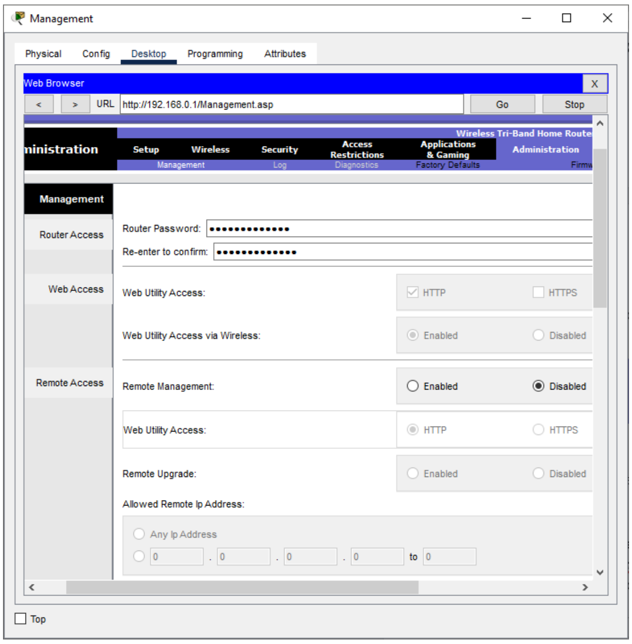


Fig.4. Saving Wireless Router Configurations

f. Re-authenticate with the new password and click Continue.

1.2 Disabling Remote Management

a. From the Administration page, select the radio button to disable remote management (Remote Management - Disabled).



URL http://192.168.0.1/Management.asp

Router Password: ••••••••••••••

Re-enter to confirm: ••••••••••••••

Web Utility Access: ☑ HTTP ☐ HTTPS

Web Utility Access via Wireless: ● Enabled ○ Disabled

Remote Management: ○ Enabled ● Disabled

Web Utility Access: ● HTTP ○ HTTPS

Remote Upgrade: ○ Enabled ○ Disabled

Allowed Remote Ip Address:

○ Any Ip Address

○ 0 . 0 . 0 . 0 to 0

Fig.5. Disabling Remote Management

b. Save the configuration changes by clicking the "Save Settings" button at the bottom of the page. This configuration change will cause the router to reset. After the restart, you can reconnect the PC to the management interface of the wireless router.

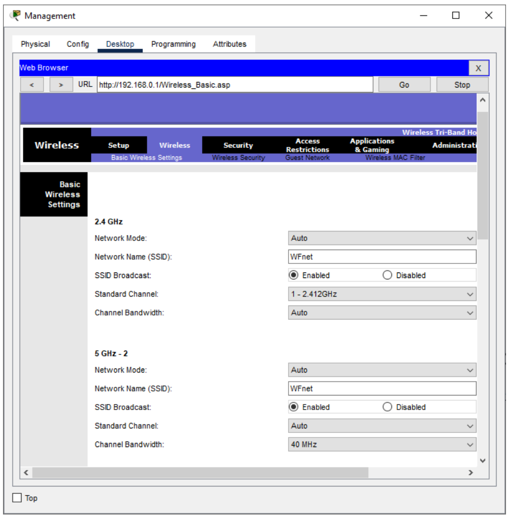
Partea 2. Configuring Network, Router, and Wireless Client Security

2.1 Configuring and Broadcasting the Wireless Network Identifier (SSID)

Some practices recommend disabling SSID broadcasting for increased security and a reduction in exposure to attacks. However, this can be inconvenient for legitimate users, and certain devices may struggle to connect to or may not connect at all to a hidden network. Additionally, the act of hiding the SSID by the administrator may serve as a warning sign for an attacker who becomes curious about why it's hidden, potentially making the network an interesting target. There are methods by which the SSID can be discovered through traffic capture and packet analysis.

a. From the Wireless - Basic wireless settings page, select the Enabled radio button (if not already selected) for Broadcast SSID, for both 2.4 and 5 GHz wireless networks.

b. For each wireless network, change the SSID identifier from Default to WFnet.



URL http://192.168.0.1/Wireless_Basic.asp

2.4 GHz

Network Mode: Auto

Network Name (SSID): WFnet

SSID Broadcast: ● Enabled ○ Disabled

Standard Channel: 1 - 2.412GHz

Channel Bandwidth: Auto

5 GHz - 2

Network Mode: Auto

Network Name (SSID): WFnet

SSID Broadcast: ● Enabled ○ Disabled

Standard Channel: Auto

Channel Bandwidth: 40 MHz

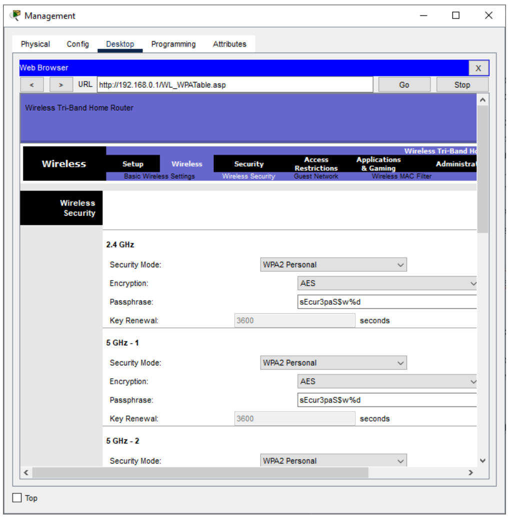
Fig. 6. Configuring and Enabling SSID Broadcasting

c. Save the changes by clicking the "Save Settings" button.

2.2 Configuring the Security Protocol for the WFnet Wireless Network

One of the most critical security measures, aside from changing the administration password and disabling remote access, is encrypting the traffic between the wireless router and clients to prevent situations where an attacker intercepts network communication and gains unauthorized access to information. There are free and easily obtainable online tools that can exploit unencrypted network traffic.

a. From the Wireless - Wireless Security page, configure the following parameters for all wireless networks:
   - Security Mode: ***WPA2 Personal***
   - Encryption: ***AES***
   - Passphrase: ***sEcur3paS$w%d***



URL http://192.168.0.1/WL_WPATable.asp

2.4 GHz

Security Mode: WPA2 Personal

Encryption: AES

Passphrase: sEcur3paS$w%d

Key Renewal: 3600 seconds

5 GHz - 1

Security Mode: WPA2 Personal

Encryption: AES

Passphrase: sEcur3paS$w%d

Key Renewal: 3600 seconds

5 GHz - 2

Security Mode: WPA2 Personal

Fig. 7. Configuring Encryption for Wireless Network Traffic

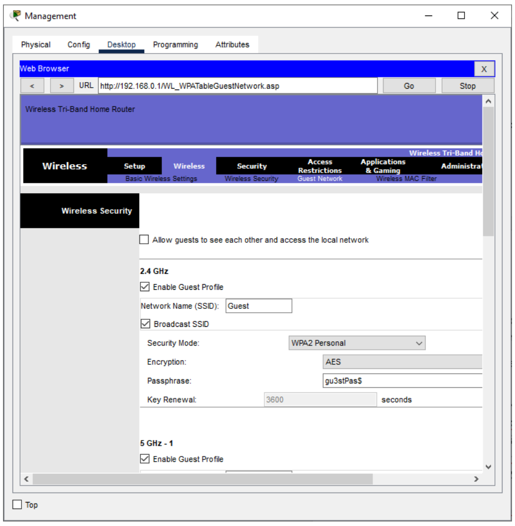
b. Save the changes by clicking the "Save Settings" button.

2.3 Configuring Security for the Guest Wireless Network

For wireless routers that allow the configuration of independent networks on each radio frequency, it is possible to separate guest traffic from that of regular users (family, employees, etc.)

a. From the Wireless - Wireless Security page, activate the guest profile by checking the Enable Guest Profile checkbox

b. Configure the following parameters for all wireless networks:
   - SSID: ***Guest***
   - Broadcast SSID***: Enabled***
   - Security Mode: ***WPA2 Personal***
   - Encryption: ***AES***
   - Passphrase: ***gu3stPas$***



URL http://192.168.0.1/WL_WPATableGuestNetwork.asp

□ Allow guests to see each other and access the local network

2.4 GHz

☑ Enable Guest Profile

Network Name (SSID): Guest

☑ Broadcast SSID

Security Mode: WPA2 Personal

Encryption: AES

Passphrase: gu3stPas$

Key Renewal: 3600 seconds

5 GHz - 1

☑ Enable Guest Profile

Fig. 8. Configuring the Guest Wireless Network

c. Save the changes by clicking the "Save Settings" button.

2.4 Configuring Wireless Clients (Laptop and Tablet)

a. Open the laptop configuration window (left-click on the laptop icon) and access the PC Wireless application from the Desktop panel.

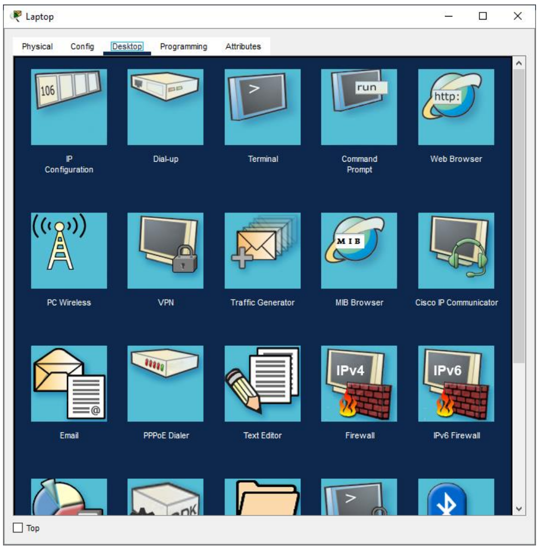
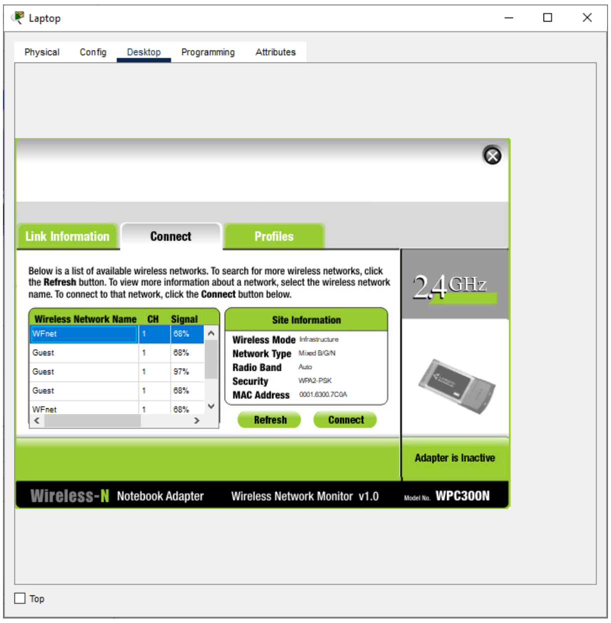


Fig. 9. Launching the PC Wireless Application from the Desktop

b. From the Connect panel, select the WFnet network and click the Connect button. Note: If necessary, click the Refresh button first.



Below is a list of available wireless networks. To search for more wireless networks, click the **Refresh** button. To view more information about a network, select the wireless network name. To connect to that network, click the **Connect** button below.

Wireless Network Name	CH	Signal
WFnet	1	68%
Guest	1	68%
Guest	1	97%
Guest	1	68%
WFnet	1	68%

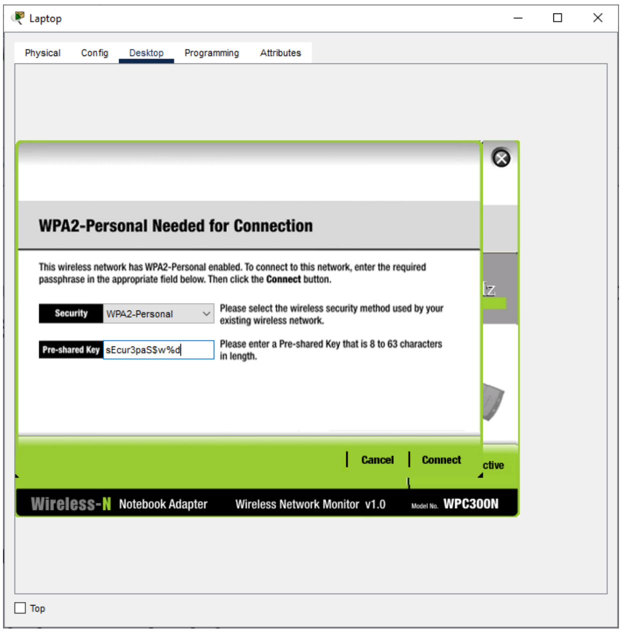
Site Information

Wireless Mode	Infrastructure
Network Type	Mixed B/G/N
Radio Band	Auto
Security	WPA2-PSK
MAC Address	0001.6300.7C0A

Adapter is Inactive

Fig. 10. Connecting to the Wireless Network

c. The security protocol is already selected (WPA2 Personal). Enter the established password (de ex. ***sEcur3paS$w%d***) in the Pre-shared key field and click Connect.



WPA2-Personal Needed for Connection

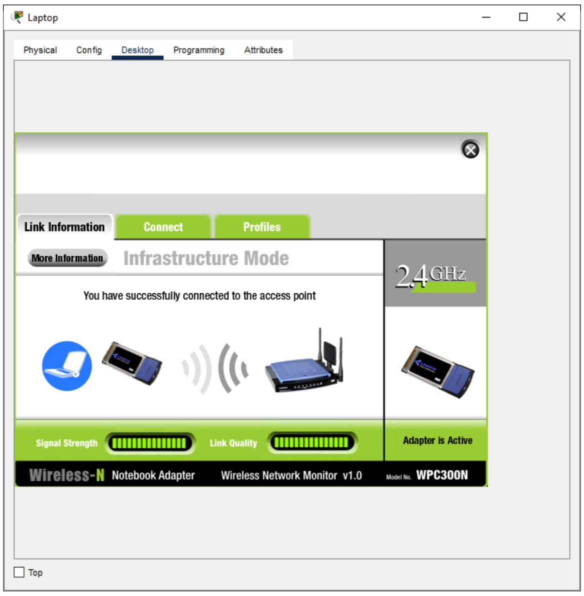
This wireless network has WPA2-Personal enabled. To connect to this network, enter the required passphrase in the appropriate field below. Then click the **Connect** button.

Security: WPA2-Personal — Please select the wireless security method used by your existing wireless network.

Pre-shared Key: sEcur3paS$w%d — Please enter a Pre-shared Key that is 8 to 63 characters in length.

Fig. 11. Entering the Key (Password)

d. In the Link Information panel, you can observe that the connection to the wireless access point has been successfully established. If this is not mentioned, repeat the previous steps and carefully enter the password.



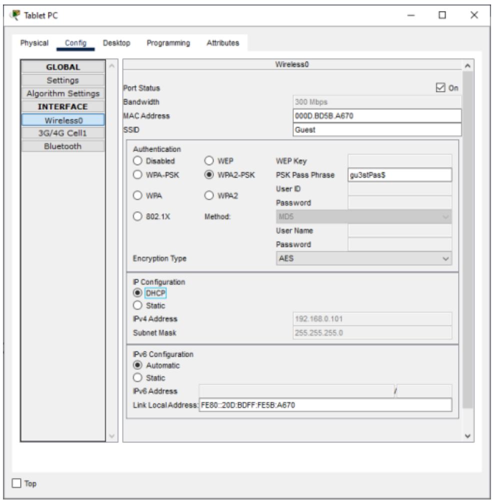
You have successfully connected to the access point

Fig. 12. Confirming the Connection

e. Close the PC Wireless application and open the IP Configuration application from the Desktop to check the IP configuration. When DHCP is selected, the laptop should receive an address from the 192.168.0.0/24 segment.

f. Open the tablet configuration window (left-click on the tablet icon), and from the Config panel, access Wireless0 from the left menu.

g. Use the parameters from the previous steps to configure the tablet's wireless connection to the Guest network.



Wireless0

Port Status ☑ On

Bandwidth 300 Mbps

MAC Address 000D.BD5B.A670

SSID Guest

Authentication

○ Disabled ○ WEP WEP Key

○ WPA-PSK ● WPA2-PSK PSK Pass Phrase gu3stPas$

○ WPA ○ WPA2 User ID / Password

○ 802.1X Method: MD5 User Name / Password

Encryption Type AES

IP Configuration

● DHCP

○ Static

IPv4 Address 192.168.0.101

Subnet Mask 255.255.255.0

IPv6 Configuration

● Automatic

○ Static

IPv6 Address

Link Local Address: FE80::20D:BDFF:FE5B:A670

Fig. 12. Configuring the Tablet's Wireless Connection

h. Check the IP configuration after the Port Status box is disabled and re-enabled, or when the DHCP radio button is selected/re-selected.

Part 3. Checking Wireless Connectivity and Security

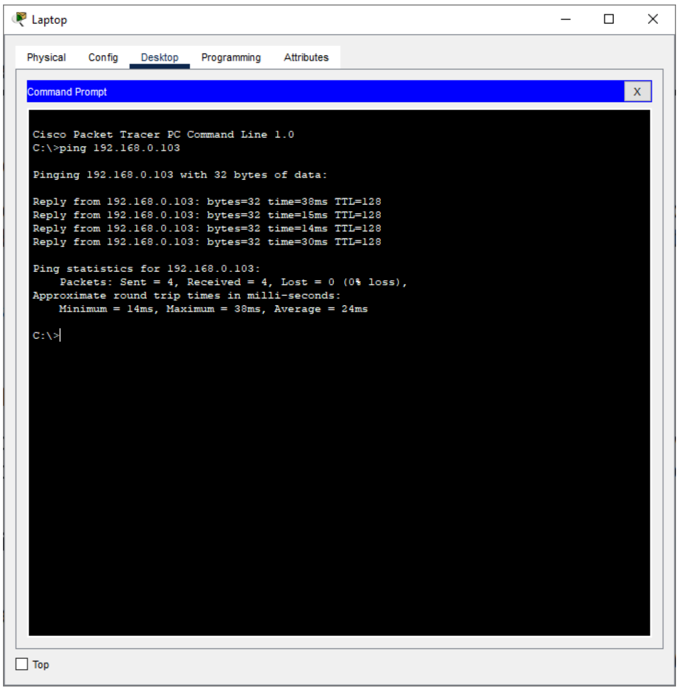
After implementing security configurations, tests should be conducted to verify whether devices can communicate with each other and if the configurations are functioning correctly.

3.1 Testing Connectivity for Laptop and Tablet

a. Open the management computer configuration window (left-click on the computer icon), and from the Desktop panel, access the IP Configuration application to find its IP address. This address should be from the 192.168.0.0/24 segment (e.g., 192.168.0.103).

b. Open the laptop configuration window (left-click on the laptop icon), and from the Desktop panel, access the Command Prompt application.

c. Enter the ping command followed by the IP address of the management computer (e.g., ping 192.168.0.103) and monitor the received messages. Its response confirms that connectivity is established.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:

Reply from 192.168.0.103: bytes=32 time=38ms TTL=128
Reply from 192.168.0.103: bytes=32 time=15ms TTL=128
Reply from 192.168.0.103: bytes=32 time=14ms TTL=128
Reply from 192.168.0.103: bytes=32 time=30ms TTL=128

Ping statistics for 192.168.0.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 38ms, Average = 24ms

C:\>
```

Fig. 13. Checking Connectivity for the Laptop and the WFnet Network

Note: If no response is received from the management computer, repeat the device configuration steps.

d. Repeat steps b and c for the tablet. What response is received? What could be the explanation?

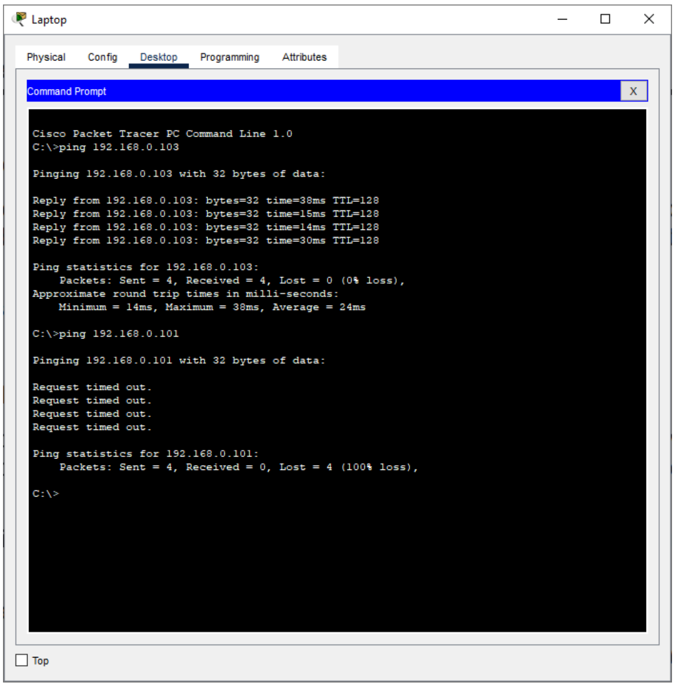
3.2 Configuring Security Parameters for Interconnecting Wireless Networks

Normally, the two wireless networks, WFnet and Guest, should not connect and share resources. The same should happen between the local network and the Guest wireless network, which is isolated from the rest of the networks.

a. Find the IP addresses of the tablet and the laptop (e.g., 192.168.0.100 and 192.168.0.101, respectively).

b. Open the laptop configuration window (left-click on the laptop icon), and from the Desktop panel, access the Command Prompt application.

c. Enter the ping command followed by the IP address of the management computer (e.g., ping 192.168.0.101) and monitor the received messages. Its response (Request timed out) confirms that the connection cannot be established.



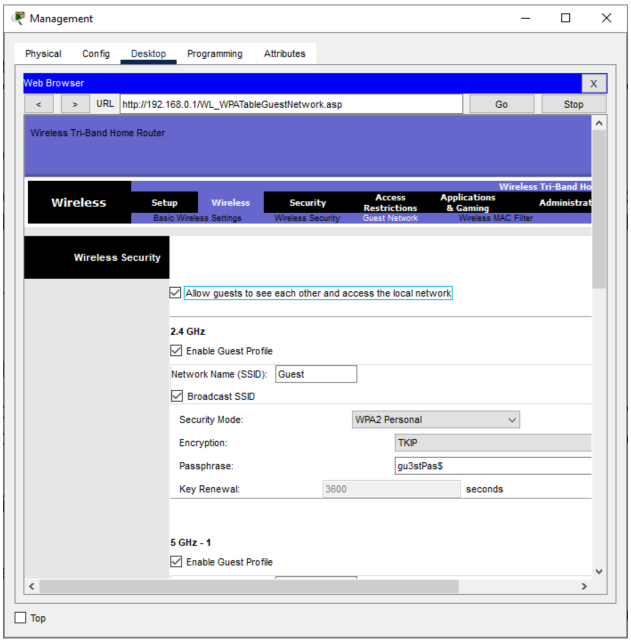
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:

Reply from 192.168.0.103: bytes=32 time=38ms TTL=128
Reply from 192.168.0.103: bytes=32 time=15ms TTL=128
Reply from 192.168.0.103: bytes=32 time=14ms TTL=128
Reply from 192.168.0.103: bytes=32 time=30ms TTL=128

Ping statistics for 192.168.0.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 38ms, Average = 24ms

C:\>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

Fig. 14. Checking Connectivity Between Laptop and Tablet

d. Repeat the previous step, trying to check connectivity from the tablet to the laptop.

e. From the management computer, connect to the web application for managing the wireless router (Desktop - Web Browser).

f. On the Wireless - Guest Network page, check the Allow guest to see each other and access local network checkbox.



URL http://192.168.0.1/WL_WPATableGuestNetwork.asp

☑ Allow guests to see each other and access the local network

2.4 GHz

☑ Enable Guest Profile

Network Name (SSID): Guest

☑ Broadcast SSID

Security Mode: WPA2 Personal

Encryption: TKIP

Passphrase: gu3stPas$

Key Renewal: 3600 seconds

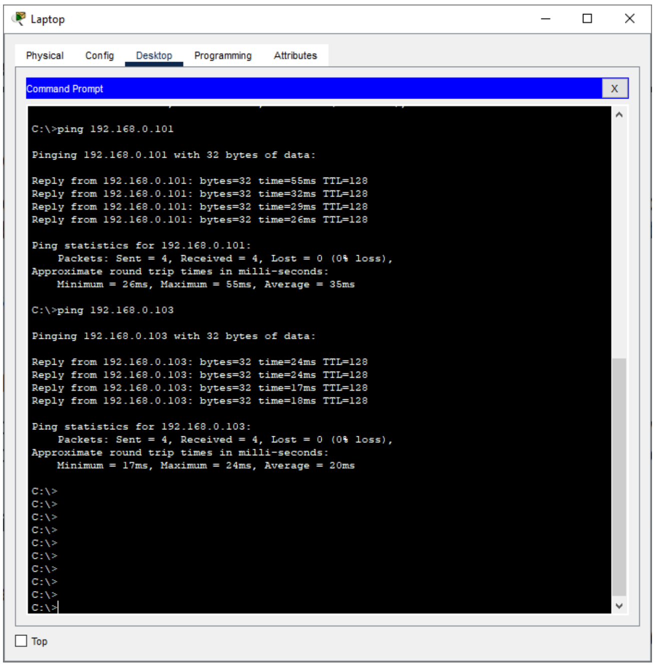
5 GHz - 1

☑ Enable Guest Profile

Fig. 15. Enabling Permission for Connection Between the Guest Network and Other Networks

g. Save the changes by clicking the "Save Settings" button.

h. Repeat steps c and d to check connectivity between the two wireless networks. Test if the management computer is accessible from both the laptop and the tablet using the same ping command.



```
C:\>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:

Reply from 192.168.0.101: bytes=32 time=55ms TTL=128
Reply from 192.168.0.101: bytes=32 time=32ms TTL=128
Reply from 192.168.0.101: bytes=32 time=29ms TTL=128
Reply from 192.168.0.101: bytes=32 time=26ms TTL=128

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 26ms, Maximum = 55ms, Average = 35ms

C:\>ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:

Reply from 192.168.0.103: bytes=32 time=24ms TTL=128
Reply from 192.168.0.103: bytes=32 time=24ms TTL=128
Reply from 192.168.0.103: bytes=32 time=17ms TTL=128
Reply from 192.168.0.103: bytes=32 time=18ms TTL=128

Ping statistics for 192.168.0.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 24ms, Average = 20ms

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
```

Fig. 16. Checking Connectivity Between the Guest Network and Other Networks

Bibliography

[1] Cybersecurity Essentials Course with Real-World Scenarios (skillsforall.com) https://skillsforall.com/course/cybersecurity-essentials?userLang=en-US

[2] Create and use strong passwords - Microsoft Support https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb