From Power to Water: Dissecting SCADA Networks Across Different Critical Infrastructures

Neil Ortiz[1][0000−0001−9008−7717], Martin Rosso[2], Emmanuele Zambon[2][0000−0002−8079−4087], Jerry den Hartog[2][0009−0001−3846−9045], and Alvaro A. Cardenas[1][0000−0002−5142−9750]

[1] University of California, Santa Cruz, USA
{nortizsi,alvaro.cardenas}@ucsc.edu
[2] Eindhoven University of Technology , Netherlands
{m.j.rosso,e.zambon.n.mazzocato,j.d.hartog}@tue.nl

Abstract. In recent years, there has been an increasing need to understand the SCADA networks that oversee our essential infrastructures. While previous studies have focused on networks in a single sector, few have taken a comparative approach across multiple critical infrastructures. This paper dissects operational SCADA networks of three essential services: power grids, gas distribution, and water treatment systems. Our analysis reveals some distinct and shared behaviors of these networks, shedding light on their operation and network configuration.
Our findings challenge some of the previous perceptions about the uniformity of SCADA networks and emphasize the need for specialized approaches tailored to each critical infrastructure. With this research, we pave the way for better network characterization for cybersecurity measures and more robust designs in intrusion detection systems.

Keywords: SCADA traffic · Network measurement · ICS · Modbus/TCP · IEC 60870-5-104

1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems represent the technology used to monitor and control remote large-scale physical processes such as the power grid, gas distribution, and water treatment. These systems manage several Critical Infrastructures so vital to society that their incapacitation or malfunction could have a debilitating effect on national security, the economy, and public health.

Despite their criticality to our modern way of life, these networks have received limited attention from the academic measurement community. SCADA systems have migrated from serial communications to IP-based and Ethernet networks in the past two decades, and they can be analyzed with the same tools we have developed for measuring other modern networks.

Most previous network measurement studies of SCADA networks focus on a single network using a single industrial protocol. As a result, they study real-world SCADA networks (and protocols) in isolation, one at a time, rather than as a unified whole. Consequently, generalizability of these studies is limited by the

scope, physical process, network layout, and used network protocols, i.e., does not describe SCADA as a whole. By studying real-world network traffic from three different physical processes, we create an overview of similarities and differences between SCADA systems. We highlight how implementation decisions and observations made on the network traffic are directly caused by the physical process and context it resides in. Our results can help putting network measurement studies of SCADA networks into context. Further, understanding the core differences and similarities between different operational SCADA systems is necessary to create better cyber security solutions (e.g., better best-practices and Intrusion Detection Systems for SCADA systems).

2 Related Work

While there is a growing interest in analyzing SCADA networks, previous measurement studies fall into two categories: (1) Use of emulated/simulated networks, i.e., testbed or laboratory environment. (2) Studies based on a single (real-world) network or single SCADA protocol.

Due to the difficulty of obtaining real-world data from operational systems, analyzing emulated or simulated data is the most popular approach [17, 12, 19]. However, simulations or testbeds do not represent the complexity and behavior of complex real-world systems. In fact, datasets emulating the power grid are prone to simple and regular patterns [16, 20]. Therefore, in this paper, we exclusively focus on data obtained from real-world operational SCADA systems, i.e., systems that monitor and control physical processes that deliver critical services for a large population.

Some papers study operational (real-world) SCADA networks, but they do not provide details of the system under study. For example, Yang *et al.* [24] captured network traffic data from a real-world SCADA system without adding details of the type of system they analyzed. Similarly Hoyos *et al.* [14] and Wressnegger *et al.* [23] indicate that their dataset comes from a power plant network, but they do not specify which network protocols are used or add any details of the network topology. Likewise, Jung *et al.* [15] analyze the TCP connections of a power distribution network without specification of protocols.

The works most closely related to ours are the 2022 PAM publication by Mehner *et al.* [20], the 2020 IMC publication by Mai *et al.* [18], and the Sigmetrics 2017 publication by Formby *et al.* [9].

Mehner *et al.* conducted a network characterization study in a Distributed Control System. They examined packet traffic at the network layer, focusing on the field, control, and HMI levels. At the field and control levels, most traffic was from a proprietary protocol, while at the supervisory level, there was no legacy ICS protocol.

Mai *et al.* conducted an analysis of IEC 104 traffic from a real-world bulk power grid. They characterized traffic at the network, transport, and application level, including topology configuration, TCP flows, IEC 104 message types, and measurement and control commands. Their findings showed topological changes from one year to the next, with 90% of TCP connections lasting less than one second, as well as non-standard IEC 104 packet configurations. This research

focused only on one protocol in one part of the power grid, whereas our study covers a wide range of protocols in all parts of the power grid.

Finally, Formby *et al.* conducted a flow analysis of substation networks using the DNP3 industrial protocol, focusing on the timing analysis of TCP connections.

While these three papers discuss a real-world SCADA system in detail, they focus only on one infrastructure. In contrast, our paper focuses on three different infrastructures (power, gas, and water) and tries to find the similarities and differences of SCADA networks under different operating conditions.

3 Background

3.1 Power Grid

Electricity grids are the foundation for generating, transmitting, distributing, and providing electricity to end-users. These systems are divided into four interconnected grids: generation, transmission, distribution, and end-consumer. Generation plants are connected to the transmission grid through high-voltage substations and transmission lines (usually rated at 220 kV and above). The combination of generation plants and the transmission grid makes up the Bulk Power System, which typically covers a large geographical area, such as an entire country. Transformer stations step voltage down, often around 50kV for the distribution network. Constant monitoring and control actions by (human) grid operators is necessary, often from a remote control room. The main activities of grid operators include monitoring, coordination of power production, import and export, as well as load balancing intervention to mange offer and demand and ensure network and frequency stability. The power grid, especially the bulk power grid, has a redundant configuration to ensure resilience against unexpected events. This paper focuses on the Bulk Power grid, and the associated central control room monitoring and remote-controlling different substations spread geographically hundreds of miles apart to get the big picture of the operation of the power grid.

3.2 Gas Distribution

The gas transport and distribution grid follows a similar structure as the power grid. At specific locations, ingest stations receive and de-pressurize gas from the high-pressure nationwide transport grid (usually $>= 50\,bar$) and inject it into medium-pressure regional transport networks (around $10\,bar$) and finally the local distribution grid. Local distribution grids operate at the level of cities or metropolitan areas. Small gas distribution closets can be found within neighborhoods at regular intervals. These stations contain a mechanical pressure regulator that decreases pressure (often to $>= 1\,bar$) for the last mile. Usually, residential consumers are connected to more than one distribution station to avoid service interruptions in case of unexpected faults or maintenance.

Similar to power distribution, gas distribution is considered part of a nation's critical infrastructure. However, gas distribution does not require consistent supervision or operator control, as gas "just" flows to where it is consumed, as

long as there is sufficient supply. As a result, there is a smaller need for redundancy, supervision, and digital control equipment. Nonetheless, stations are usually equipped with network connections to allow the operator to remotely monitor their system.

3.3 Water Treatment

Unlike the Power and Gas distribution networks that span a wide geographical area, water treatment plants are confined to a single, often relatively small, facility. Facilities highly differ based on the supply needs, ranging from a few thousand square feet for a few hundred thousand gallons per day in small communities to several acres for millions of gallons per day for cities or industrial complexes. The purpose of these plants is to remove contaminants and pathogens from water to make it safe for drinking, use in industrial processes or for safe return the natural water cycle. The treatment process varies depending on the source of the water (natural sources like reservoirs or wastewater), the type of expected contaminants (e.g., based on geographical region), and its intended use. It can be drinking water, wastewater, or a water recycling facility.

Treatment processes can be divided in three major types. (1) Physical treatment involve separating pollutants by physical characteristics such as weight (sedimentation) or size (filtration) including e.g., micro-filtration or reverse osmosis. (2) Biological treatment uses microorganisms to metabolize pollutants and convert them into biomass that can be physically removed by settling and filtration. Coagulants aid in forming solid clumps in water (coagulation), which settle as sludge (sedimentation) and are filtered out. (3) Chemical treatment purifies water by adding chemical substances, such as chlorine or ozone, to inactive pathogens (disinfection). Ozone precedes filtration, while chlorine follows to ensure the elimination of any lingering pathogens. Ultraviolet light at specific wave-lengths is used to break down cellular structures as part of the disinfection process. Individual treatment steps are prolonged or repeated until the water meets the required characteristics to move to the next treatment process.

In terms of operation technology, water treatment plants are equipped with a range of sensors and actuators that are connected to programmable logic controllers (PLCs).

PLCs are in turn connected to a supervisory control and data access (SCADA) server, acting as a centralized data concentrator for operator stations (HMIs) which allow operators to monitor the treatment process and to carry out (manual) process control when necessary. Sensors measure different parameters, such as water flow rate, turbidity, and chemical levels. Actuators include different valve types (e.g., filter flow control, aeration blower inlet, chemical addition), different damper types (e.g., for incinerator exhaust gas or incinerator fan) as well as pump motors. The PLCs are responsible for running the water treatment logic by changing valve positions, controlling pump speeds, or adding chemicals based on the different sensor readings, as well as monitoring safety conditions and raising process alerts. Process data and alarms is collected by the SCADA server and shown to Operators that monitor its correct execution and can carry out manual process control if needed.

3.4 Datasets

Our SCADA traffic dataset consists of network packet captures from three different infrastructures. Our analysis is focused on the primary SCADA protocol, i.e., the protocol used to monitor and control the respective physical process. For the power grid and gas distribution dataset, this is IEC 104, a SCADA protocol initially developed for electrical engineering and power system automation. For the water treatment process, the primary SCADA protocol is Modbus/TCP, a commonly used, non-proprietary SCADA protocol. A brief summary of the datasets is depicted in Table 1. In the remainder of this section we briefly describe each dataset and the modalities of how it was captured.

Table 1: Summary characteristics of the datasets.

ICS	Protocol	# hosts	Duration (hours)	# Packets	Name
Power	IEC 104	42	8.2	7.1 M	Power
Gas	IEC 104	157	2037	86 M	Gas
Water	Modbus/TCP	100	24.5	71 M	Water

Power Grid The network capture was collected at the border router of the SCADA control room of an Independent System Operator (ISO) in the Americas, capturing traffic from the control room to the remote controlled sites (transmission and generation substations, as well as other control rooms). This ISO manages a bulk power grid serving a population of around 40 million people over a wide geographical area. The primary SCADA protocol in this dataset is IEC 104. Additionally, this dataset contains other domain-specific protocols like e.g, the Inter-Control Center Communications Protocol (ICCP), which is used for information exchange with other control rooms. Using IEC 104, the ISO's Energy Management System monitors and controls generation plants, e.g., by using Automatic Gain Control (AGC). The control room also monitors frequency at a few transmission substations. Lastly, the control room indirectly monitors (via ICCP) other transmission substations directly connected to other control rooms. For the purposes of this study, we will focus on analyzing the traffic related to the IEC 104 protocol. This network consists of 4 control servers and 38 RTUs (IEC 104 outstations) on 23 remote sites. Each remote site is either an electric generator (17) injecting power into the grid, or transformer (6) to connect medium- or low-Voltage networks to the high-Voltage bulk power grid.

Gas Distribution The network capture was collected at a core switch inside the control center of a local distribution network operator of a Europen metropolitan area. The primary SCADA protocol in this dataset is IEC 104, used by the SCADA system to collect data and control 155 geographically spread Remote Terminal Units (RTUs). The capture also contains internal communication

among devices in the control center (among others, typical IT network protocols like DHCP, DNS, SMB, Telnet, ... and proprietary protocols, including Oracle TNS and two unknown and probably vendor specific ICS protocols between ICS services). Most RTUs are installed at "distribution stations", however, a small number of RTUs control and monitor a gas turbine, biogas generation and injection, and testing equipment.

Water Treatment The network capture was collected at a core switch of a large (approx. 800 square meters) wastewater treatment plant. The facility collects wastewater from a population of approximately 650 thousands people in an American city and, after treatment, releases the purified water into a river. The control system comprises field devices (40 mid/large size PLCs, 18 small size PLCs and other 16 field devices, mostly serial-to-Ethernet gateways), two redundant SCADA servers in hot standby configuration, two application servers, one historian server, one engineering workstation and 15 operator stations. The primary SCADA protocol in this dataset is Modbus/TCP, with the primary SCADA server being the only one actively communicating with field devices. ICS endpoints in the control center (SCADA servers, application servers, historian servers, and the operator stations) communicate with each other using standard IT protocols (SMB, Kerberos, DNS, etc.) and two proprietary protocols of the General Electric iFIX SCADA software suite. The ICS network segment is mostly flat, with endpoints from both the control center and field devices belonging to the same collision domain.

4 Analysis

For simplicity, in the remainder of this paper, any endpoint that reports data will be referred to as an **agent (A)**, while any endpoint that collects data will be referred to as a **controller (C)**. For example: we will consider any PLCs or RTUs as "agents" and HMI or SCADA server as "controllers". In this way, we can focus on the characteristics of the network to facilitate our comparisons and diagrams, and not dwell on protocol-specific naming conventions.

Figure 1 illustrates how our research questions create a general framework for analyzing SCADA networks. We start by understanding the topology of each network (RQ1), and then we start zooming in to understand the traffic differences between networks (RQ2), then traffic differences within a network (RQ3), the data types handled by the network (RQ4), and finally, the types of measurement and control commands sent back and forth between a controller and the agents (RQ5).

RQ1: What are the topology differences between the three SCADA networks?
RQ2: How are the three SCADA networks different in terms of packet size and timing?
RQ3: How are the three SCADA networks different in terms of packet size diversity?
RQ4: What type of information is carried by the ICS protocols in the three SCADA networks?

RQ5: How much monitoring vs. control is done in these networks, and what types of control commands are sent?

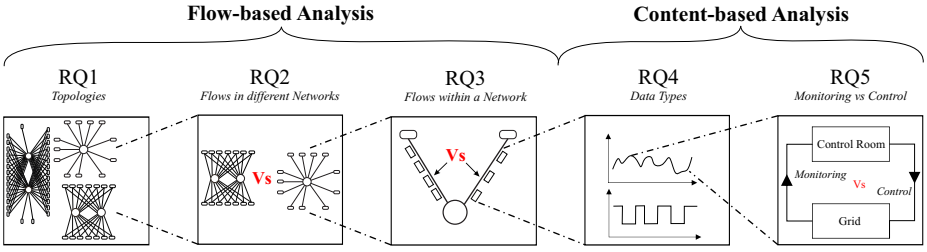


Fig. 1: Framework for our research questions.

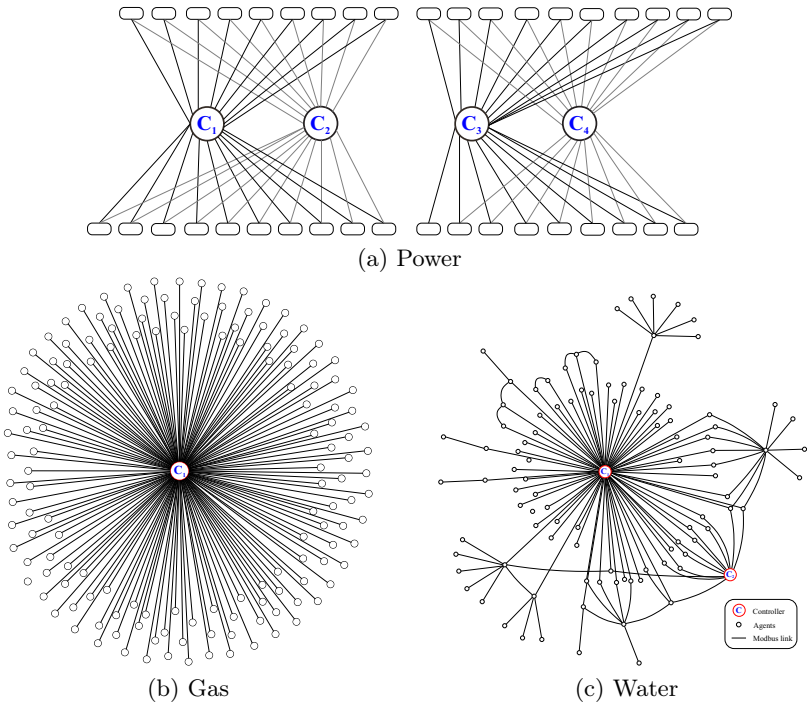
4.1 RQ1: Network Topology

Our first goal is to understand the differences in topology of the three SCADA networks, as well as reasoning about the underlying choices that induced the respective topologies. Figure 3 is a visual representation of the network topology of all three SCADA networks. Even though the network topologies seem different at first glance, all networks are constructed around one or two central controllers.

Power We observe that the power grid topology from Figure 2a forms a complete bipartite graph. A complete bipartite graph $K_{p,q}$ consists of a set of p vertex and a set of q vertex (in our case, $p = 2$) and pq edges joining the vertex of different types [13]. This type of topology is known as spine leaf topology [22, 11] in cloud data centers. The difference is that the spine leaf topology is used to forward packets through the spine (the central nodes), while in SCADA networks, the central nodes consume data (they do not forward it).

In our power network, each agent is connected to two controllers. This dual-purpose setup offers fault tolerance and load balance, which are essential for a network that focuses on the operational status of the process (see Section 4.3). This reduces the risk of the operators losing visibility in the event of a controller or link failure. If one controller fails, the other will take over, allowing control applications such as the Automatic Generation Control (AGC) algorithm to access the input data needed for its control operation. Furthermore, the operator can still monitor the grid from their HMI [4, 3].

Upon looking at the traffic of each connection, we find that from the two connections to a pair of controllers, one of these connections is used to send process data to one of the controllers (active link), while the other is used to keep the connection with the other controller alive, serving as a backup. The heartbeat signal consists of U-Format messages (TESTFR) described in the IEC 104 protocol.



(a) Power

(b) Gas

(c) Water

Fig. 2: Type of topology structure: (a) Complete bipartite, (b) Star, (c) Star-Hybrid: a star topology with other structures.

This type of load balancing makes sense as we only require one active connection between a controller and an agent while the other connection is on standby, ready to be used in case of a failure.

In summary, we see two bipartite graphs: $K_{2,18}$ and $K_{2,14}$. We further confirm with the ISO that all four controllers are physically in the same control room, so this network represents a control room with four servers arranged in pairs, each monitoring a different part of the grid. In contrast to the other topologies, we believe these $K_{2,q}$ graphs represent networks that are more critical, such as networks for control operations, than the simple star graphs and, therefore, need standby connections to the controllers.

Gas The gas network exhibits a star topology, with a controller connected to 155 agents, as illustrated in Figure 2b. By number of agents, this is the largest network in our dataset.

Contrary to the IEC 104 standard that requires agents to have a second fail over TCP connection to a backup control server open at all times, agents in this gas distribution network do not open a secondary TCP connection. However, a second control server is present in the network, as we can see agents switching

to a different control server during a network maintenance period recorded in our dataset.

The observed star topology can be explained by two considerations. First, some form of redundancy is provided at the network level (e.g., configured on network switches), that is not captured by our IP topology. If there was no additional fail-safe mechanism, an outage of the single control server would render the control room inoperable. Secondly, the distributed control structure of the gas distribution network does not require constant supervision from a central control server. Instead, agents run a local control loop that ensures the correct execution of the process. Thus, availability of the control server is not essential for the execution of the physical process and short interruptions "only" inhibit monitoring of the current process state.

Water Finally, the water network is characterized by a star-hybrid topology, as seen in Figure 2c.

We see that this network has the same single point of failure as the gas network; however, we also notice that the endpoints collaborate and exchange data at the edge of the network.

One major difference between our water network and the power and gas networks is that the water network is a LAN, and the gas and power networks are WANs. We cannot see the MAC addresses of remote devices in the power and gas networks; however, in the water network, we can see them and we can identify them as mainly Programmable Logic Controllers (PLCs). In addition, a LAN network suggests that the water treatment facility is not spread over a large physical area, all devices are setup in relatively close proximity. It can be assumed that the water treatment plant prioritizes operational reliability and simplicity.

Discussion Network topology is shaped by the needs of the physical process under control. Managing offer and demand in the power grid requires constant and timely interventions from a centralized control room. Gas distribution, on the other hand, is a distributed process that agents can execute without constant connection to the control room. As a result, redundancy and fail-over techniques are more sophisticated in the power grid. Each agent is always connected to the primary control server *and* a backup control server. By spreading agents over multiple control servers, the impact of a failing control server on the physical process is limited. This is not necessary for gas distribution and may even result in unnecessary complexity. Instead, the topology of the gas network is kept simple.

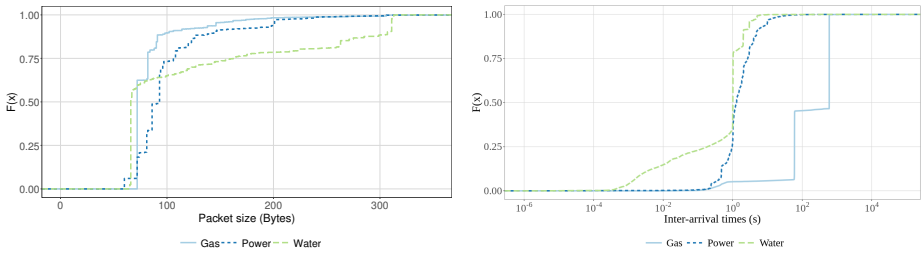
The topology of the water treatment plant shows that most agents communicate directly with the control server. However, it also shows agents communicating among each other, as well as sub-controllers connecting multiple agents indirectly to the main controller.

The water purification process is relatively complex and divided into subprocesses. Often, it is enough to know and control the state of the process (e.g., water draining) and have the different agents take care of driving the actuators according to the desired state.

4.2 RQ2: Packet Size and Timing

We now turn our attention to the traffic flows in these networks, to identify if and how they differ. The classical traffic analysis metrics are packet sizes and the timing (or inter-arrival times) of transmissions.

Figure 3a and Figure 3b visualize the distribution of packet sizes in bytes and the inter-arrival time between two packets for all three datasets.



(a) Packet sizes (b) Inter-arrival time

Fig. 3: CDF for (a) packet sizes and (a) IAT for each ICS network.

Power The majority of packets (75%) in the IEC 104 traffic of the power grid dataset are smaller or equal to 100 bytes. Almost all (99.99%) of all observed packets are shorter than 200 bytes. The largest observed network packet was 1378 bytes, combining multiple IEC 104 messages in a single TCP segment.

The distribution of packet inter-arrival times (see Figure 3b) shows no discrete steps or plateaus. This implies that the agents do not have a regular reporting pattern but instead that their communication pattern constantly changes. This is a result of the many spontaneous messages of this network, i.e., messages that an agent automatically sends to the control server when a value or status changes. The continuous yet varied slope in Figure 3b implies that most messages are spontaneous or event-driven. This is a sign that the network constantly responds to the dynamic conditions of the grid and the power delivery process.

Gas The vast majority (80%) of messages in the IEC 104 traffic of the gas distribution dataset are 100 bytes or less, with 99.99% of the messages being shorter than 200 bytes. Figure 3a indicates that most packets ($\sim$65%) are $\sim$70 bytes, which indicates a limited variety of message types being predominantly sent over the network. The largest packet observed (744 bytes) is roughly half the length of the largest packet observed in the power grid dataset and is also here combining multiple IEC 104 messages in a single TCP segment.

When looking at packet frequency, the gas distribution network is the least active. The transmission rate of network packets is one order of magnitude lower than for the other two SCADA networks (see Figure 3b). In fact, 94% of all

agents in the gas distribution network transmit IEC 104 packets in the order of minutes, indicating that the controller does not need fast updates, because process control operations are automatically done at the substation (i.e., the agent) with a local control program.

Figure 3b shows several clearly distinct steps: less than 1 second (5.5%), 1 minute (40%), and 10 minutes (54.5%). These steps suggest that, for most agents, there is a predefined schedule based on which messages are sent.

Water For the water treatment dataset, 65% of all messages are shorter or equal to 100 bytes. In contrast to the power and gas systems, we observe a significantly larger amount of packets in the 100-200 bytes range.

In Figure 3b, we observe that inter-arrival times begin at the milliseconds rates. However, this is the case only for a small fraction of data points (less than 1%). A quarter of all network devices have a transmission rate interval of less than a second.

The controller interrogates 43% of the agents every second (there are no spontaneous or agent-initiated messages in the Modbus/TCP protocol, so all message exchanges are the response to a query from the controller). There is a degree of consistency in the configuration of the devices in the Modbus/TCP protocol.

Discussion For all three networks, most of the SCADA protocol messages are shorter than 100 bytes. This is not surprising, as SCADA protocols usually do not transfer large amounts of variable length (textual) content: setpoints, sensor readings, commands, status changes, etc., are all fixed, and relatively short length, binary encoded, messages.

The high difference in packet lengths (see Figure 3a) can be explained by the differences between the IEC 104 and Modbus/TCP protocols. A Modbus/TCP message can encapsulate a large number of (contiguous) registers in a single packet. IEC 104 packets also include overhead for the address of data points and typically need to divide large responses into multiple messages. For example, a Gas network agent might respond to an Interrogation command with 105 status data points spread across five packets, while Modbus/TCP would consolidate the same amount of data into a single reply, or even 2000 status data points in a single packet as seen in the biggest packet size (313 bytes).

The notably high messaging intervals in the gas distribution network reflects the distributed control strategy of the process. While the power grid is centrally managed and controlled, in the gas distribution grid each agent runs a local control program and centralized control is not necessary. Thus, the gas distribution control center does not need real-time information about the current state of the grid. For the water treatment process, although process control happens locally at the agents (similar to what happens in gas distribution), the agents are local to the control center and frequent data polling is "cheap", giving plant operators a more real-time view of the process being monitored.

In summary, we can see some similarities and differences among these networks. Further, we observe similarities and differences in terms of message intervals, caused by different monitoring strategies. In Modbus/TCP, agents do not

act or send data independently, but instead monitoring data has to be actively polled by the controller in an interrogation process. In contrast, IEC 104 has the ability to configure agents to automatically report whenever a tracked value exceeds a threshold (the standard calls this "spontaneous"). As we discuss later, our power grid network takes advantage of this, and therefore, the transmission patterns are more diverse. Even though the gas network also uses IEC 104, it does not utilize this feature but instead is configured so that the controller explicitly interrogates the agents in regular intervals.

This difference in monitoring philospohy also explains difference in our packet size analysis. Almost half of the packets in both the Gas network and the Water network are interrogation queries, thus these packets have the same size. Every interrogation query requires a response from an agent. As the response contains a process variables, it is larger than the interrogation packet. This explains the big step in Figure 3a for gas and water. For the power grid network, however, there are no repeated interrogation queries (that would have the same packet size). Because the distribution of message types is more diverse and spontaneous messages can push monitoring data without request, we observe a greater variety of packet sizes in the power grid.

Finally, we also observe that the power and water networks have shorter transmission times (for different reasons). Having said that, since both IEC 104 networks operate in a WAN, the minimum inter-arrival time in our datasets for any of them is 100ms. This may identify a time constraint for these networks managing assets in large geographical areas.

4.3 RQ3: Packet Size Diversity

We now look at the diversity within networks. Entropy is a helpful metric for understanding the diversity (in terms of randomness) of data, and we can use it to evaluate the randomness of packet sizes within a network.

In the context of packet sizes, a higher entropy value (values close to 1) suggests a wide range of packet sizes being sent/received by an agent, while a lower entropy (closer to 0) implies that most packets have similar sizes. The spread and range of entropy values for each network provide insights into how varied the packet sizes are within each system. By comparing the entropy distribution using a cumulative distribution function (CDF), we determine which network has the most predictable packet size distribution.

We use the Shannon entropy formula:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log(p(x_i))$$

Where $p(x_i)$ is the probability of occurrence of a particular packet size for a particular agent. For each agent, we count the occurrence of each unique packet size. Then, we divide each count by the total number of packets for that agent to get the probability distribution. Then, we add all the packet sizes a particular agent has sent at least once.

Figure 4 shows the distribution of packet size entropy values for the three networks. In the context of this plot, higher entropy values represent greater

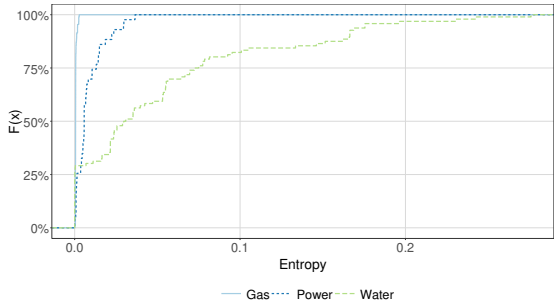
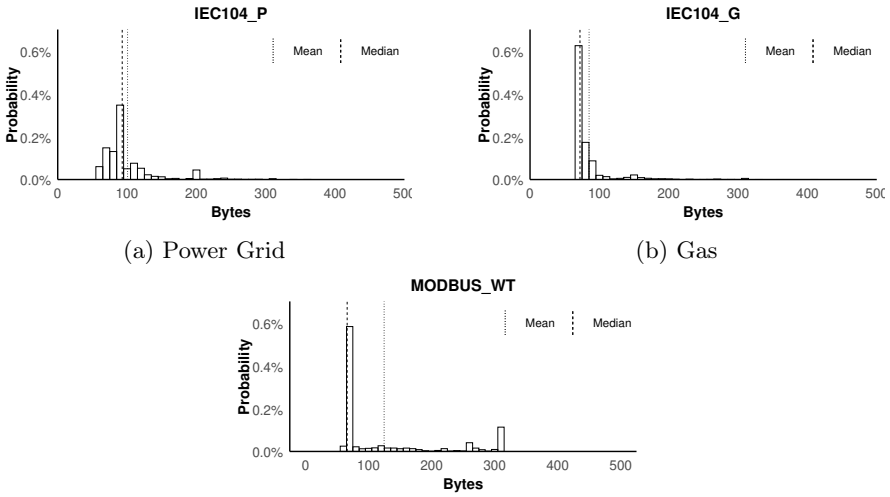


Fig. 4: CDF of packet size entropy

packet size diversity. The network with more agents (y-axis) having higher entropy values (x-axis) and more variability in its curve is likely the network that is less predictable in terms of packet sizes. Complementary to this view, Figure 5 shows the packet size distribution for the three datasets.



(a) Power Grid

(b) Gas

(c) Water Treatment

Fig. 5: Packet size distribution. Bin width 10.

Power From Fig.4, we see that ∼25% of the agents have an extremely low entropy value (close to 0). An additional ∼30% of the agents also have very low entropy. For the remaining agents the entropy is also low, in the range 0.02–0.04. From Figure 5b, we identify that packet sizes are primarily concentrated around 50-100 bytes. There are two modes: a primary mode in the 100-byte bin (90-100

bytes), which is the highest peak in the distribution. A secondary or minor peak at 200 bytes (exactly at 198 bytes). The former are spontaneous, and the latter are periodic (1-second) packets containing information (I-format in IEC 104).

Gas From Fig.4, we see that ∼85% of the agents have an extremely low entropy value (close to 0). For the remaining agents the entropy is also very low (below 0.01). From Figure 5b, we notice that the vast majority of packets are small. 70% of them are, in fact, S-format and U-format messages (packets defined in IEC 104 to acknowledge the receipt of data and for checking the status of a connection). While the default rate for acknowledging packets is higher in IEC 104, the gas network needs to send pretty much an S-format acknowledgment for every I-message received because of the long interval of time between transmissions.

Water From Fig.4, we see that ∼25% of the agents have an extremely low entropy value (close to 0). An additional ∼50% of the agents have an entropy below 0.1. For the remaining agents the entropy is also relatively low, in the range 0.1–0.3. From Figure 5c, we see a heavy tail of large packets. We identify that most of the small packets correspond to `Read Coil` operations (function code 1). These are binary values and usually represent the status of an actuator or the presence of a certain process or diagnostic alarm signal. In contrast, we see that the large packet sizes correspond to `Read Holding Registers` operations (function code 3).

Discussion From the entropy analysis, we can see that a quarter of agents have uniform packet sizes. The three networks show a sharp increase at first until 25% (y-axis), meaning that a large proportion of agents in the three networks have extremely low entropy values. This implies that one-fourth of their agents have a consistent packet size or lack of diversity in packet sizes. For example, the agent with the least entropy in Power, sends repeatedly the same packet size of 72 bytes.

Most of the agents in the Gas network exhibit extremely low entropy values, implying a lack of diversity or uniformity in packet sizes for these agents. Therefore, most of the packets have a fixed length.

The Water network has the highest entropy of the three, which means that there is more uncertainty about the packet size that the controller will receive.

4.4 RQ4: Information Handled by ICS Protocols

We now turn our attention to the information contained within the packets themselves. Each protocol has its own specific standard with clearly defined object model and associated data types that are included in the packets. We start with a general overview in Figure 6. On the x-axis, we see the number of data types defined by the standard's object model, and on the y-axis, we see the number of data types present in our capture. There are two clusters: on the top right corner, we see the IEC 104 networks, with over 100 data types defined in the standard's object model but only using around 10% in the capture. On

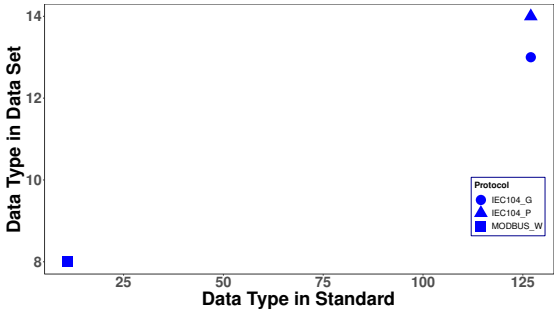


Fig. 6: Data Types.

Table 2: Power ASDU types and their description

Type	Reference	Description	%
1	M_SP_NA_1	Single-point information.	<0.001
3	M_DP_NA_1	Double-point information.	0.08
5	M_ST_NA_1	Step position information.	0.05
7	M_BO_NA_1	Bitstring of 32 bit.	<0.001
9	M_ME_NA_1	Measured value, normalized value.	1.97
13	M_ME_NC_1	Measured value, short floating point number.	39.71
30	M_SP_TB_1	Single-point information with time tag CP56Time2a.	<0.001
31	M_DP_TB_1	Double-point information with time tag CP56Time2a	<0.001
34	M_ME_TD_1	Measured value, normalized value with time tag CP56Time2a.	0.51
36	M_ME_TF_1	Measured value, short floating point number with time tag CP56Time2a.	57.21
50	C_SE_NC_1	Set point command, short floating point number.	0.41
70	M_EI_NA_1	End of initialization.	<0.001
100	C_IC_NA_1	Interrogation command.	0.019
103	C_CS_NA_1	Clock synchronization command.	0.001
135	Unknow	not defined.	0.03

the bottom left-hand corner is the Modbus/TCP network. The Modbus/TCP standard's object model does not define many data types so, obviously, the data types in the capture are fewer in absolute terms.

An IEC 104 packet can be either in Information (I), Supervisory (S), or Unnumbered (U) APCI format. I-format packets are used to exchange sensor and control data, while S and U-format packets are only used for network signaling (acknowledgments and heartbeats, respectively). For I-format packets, the standard [1] defines 127 different data types. However, in the power dataset we only count 14, and in the gas dataset 13 being used.

In contrast to IEC 104, other standards do not have the same diversity. Modbus/TCP is one of the oldest and simplest protocols used in SCADA systems. Its object model only defines three types of data: (Coils) single bit variables for binary values such as ON/OFF, (Registers) 16-bit variables for continuous values, and file records. Modbus/TCP defines 11 function codes to interact with these data types.

It is possible that we have not observed all the data types used in the power and water networks, due to the brief duration of our traffic captures. Nevertheless, since our traffic is consistent with steady-state conditions, our capture

Table 3: Gas ASDU types and their description

Type	Reference	Description	%
1	M_SP_NA_1	Single-point information.	27.85
3	M_DP_NA_1	Double-point information.	0.03
9	M_ME_NA_1	Measured value, normalized value.	17.92
30	M_SP_TB_1	Single-point information with time tag CP56Time2a.	1.13
34	M_ME_TD_1	Measured value, normalized value with time tag CP56Time2a.	11.38
45	C_SC_NA_1	Single command.	0.01
48	C_SE_NA_1	Set point command, normalized value.	0.01
58	C_SC_TA_1	Single command with time tag CP56Time2a.	<0.001
70	M_EI_NA_1	End of initialization.	<0.001
100	C_IC_NA_1	Interrogation command.	36.38
101	C_CI_NA_1	Counter interrogation command.	<0.001
102	C_RD_NA_1	Read command.	3.79
103	C_CS_NA_1	Clock synchronization command.	1.47
200	Unknown	not defined.	0.03

Table 4: Modbus/TCP types and their description

Type	Description	%
1	Read Coils	9.52
2	Read Discrete Inputs	0.39
3	Read Holding Registers	89.1
4	Read Input Registers	0.66
15	Write Multiple Coils	0.34
16	Write Multiple Registers	<0.001
22	Mask Write Register	<0.001
23	Read/Write Multiple Registers	<0.001
90	Schneider Unity	<0.001

reflects the most frequent data types in use during the operating stage of the respective processes.

Power Table 2 reports the 14 (known) information types recorded in the power dataset. We see that 99% of the messages exchanged corresponds to only two types of messages: type 36 (Measured value, short floating point value with time tag) and Type 13 (Measured value, short floating point). These data types are used for exchanging power, voltage, and current measurements. The next most frequently used message (at only 0.845%) is perhaps the most critical message sent in this network: type 50 (Set point command, short floating point number). It is by means of these setpoint command messages that the SCADA system influences the behavior of (possibly large) power generators to keep the stability of the grid under control.

Gas Table 3 reports the 13 (known) information types recorded in the gas dataset. Approximately 36% of the messages exchanged in the gas network are interrogation commands (the controller requesting information from agents), reflecting the polling-based use of IEC 104 in this setup. While the data types

in the power network are dominated by continuous values, in the gas network we see an almost equal split between discrete (28.98%) and continuous (29.3%) values. Messages communicating discrete values are in turn split between type 1 (Single-point information) and type 30 (Single-point information with time tag CP56Time2a). The difference between the two is the presence (or absence) of a timestamp next to the point value, indicating the time at which the point value was measured. Reported discrete values in the gas network include several process and diagnostic alarm signals, as well as the state of the valves in the distribution network. Messages communicating continuous values are in turn split between type 9 (Measured value, normalized value) and type 34 (Measured value, normalized value with time tag CP56Time2a). As for the discrete types, the difference between the two is the presence (or absence) of a timestamp next to the point value, indicating the time at which the point value was measured. Reported continuous values in the gas network include several measures about gas (pressure, flow, temperature), as well as the position of gas pressure reduction valves (in percentage). Apart from interrogation, counter interrogation, read and clock synchronization commands, we also observe a small percentage of messages containing commands to set the value of discrete points (types 45 and 58), and of continuous points (type 48). While the vast majority of the set commands for discrete points are used to acknowledge process and diagnostic alarms, and all the set commands for continuous ones are used to adjust the value of alarm thresholds, a minority of the discrete set commands are also used by operators to manually control the gas network (e.g., by opening/closing a specific valve) during localized maintenance.

Water Table 4 reports the 8 standard Modbus/TCP function codes recorded in the water dataset as well as the Schneider Electric proprietary function code 90. As with the power network, we can see that measuring continuous variables with function code 3 (Read Holding Registers) and function code 4 (Read Input Registers) makes up most of the Modbus/TCP messages (89.1% in total) in the water network.

While input registers contain the value of analog sensor signals acquired by the agent (e.g., a PLC), holding registers contain a variety of continuous values, from internal variables used by the PLC logic program to output signals.

Measuring discrete variables with function code 1 (Read Coils) and function code 2 (Read Discrete Inputs) is the next most prevalent type of Modbus/TCP message (9.91% in total).

While discrete inputs contain the value of binary sensor signals acquired by the agent (e.g., a PLC), coils contain a variety of discrete values, from internal variables used by the PLC logic program to output signals.

A very small percentage of the messages are control commands changing the value of discrete or continuous variables. As for the gas network, most of these commands are related to alarm management.

Discussion While the IEC 104 standard's object model defines a large amount of data types, not all of them are used at any given IEC 104 network.

Although very similar in the use of IEC 104 data types (see Figure 6), the two IEC 104 networks show some differences in use of data types. We now discuss the three most notable differences. (1) The representation chosen for continuous values is mostly floating point numbers in the power network (types 13, 36 and 50), and exclusively normalized values in the gas network (types 9, 34 and 48). Normalized IEC 104 data points only use two bytes for representing values, while floating points use four. We deduce that in the gas network engineers chose to favor space efficiency on wire over precision of transmitted values, a choice that could be possible in the gas distribution domain, where the purpose of data acquisition is for operators to oversee a process that runs autonomously, but not in the power transmission domain, where the precision of transmitted values is important for the correct functioning of the grid stability algorithms that run in the control center. (2) Differently from the power network, in the gas network the controller sends commands to set discrete points (types 45 and 58). This can be explained by the differences in the process control at the two scenarios. In the power transmission control system, the controller is responsible to ensure the stability of the grid by computing how much power each generation site should introduce in the grid at any given time, thus requiring to send messages to generation sites to set the value of continuous data points (the generation setpoint). Instead, in the gas network the process is controlled locally at each substation and the main goal of the central controller is for operators to check the process is running as expected (e.g., by processing alarm signals) and to carry out manual intervention in case of faults or maintenance. As we have seen previously, the latter two operations involve setting the value of discrete points, i.e., to acknowledge alarm signals or to manually trigger valve open/close procedures. (3) In the gas network, we observe more messages sending commands sent by the controller to explicitly interrogate agents (types 101 and 102). This can be explained by our previous observation that in the gas network, rather than leveraging the spontaneous reporting normally adopted in IEC 104 networks, the controller is configured to poll at set time intervals the value of all points exposed by the agents. This involves interrogating agents for data types that are not covered by the general interrogation command (e.g., counters) and, in some cases, explicitly issuing a read request for specific data points.

Finally, including also the water network in our analysis, we also notice that the power and water networks monitor and control mostly continuous signals, while the gas network monitors and controls an equal amount of continuous and discrete signals. This can be explained once more by the by the differences in the controlled processes.

4.5 RQ5: Monitoring vs. Control

We now address our last research question, which relates to the direction of network flows, in particular the initiator of a flow. Intuitively, we expect SCADA networks to have more messages sent from agents to the controller(s) than commands send by the controller(s) to the agents. However, we do not see this pattern in most of our networks, mostly because of interrogation commands.

We define a2C (short for $a \rightarrow C$) as the flow direction from the agent to the controller and C2a (short for $C \rightarrow a$) as the controller to the agent.

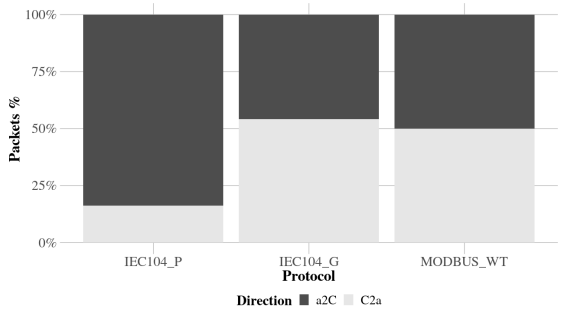


Fig. 7: Flows: Controller to agent (C2a) and agent to Controller (a2C)

Power ($a2C > C2a$) In the power dataset $a2C = 83.7\%$ and $C2a = 16.3\%$. That is, most of the packets come from the remote substations (agents) to the controller. We also make the following two observations.

Event Driven Four-fifths of the traffic is in the monitoring direction ($a2C$). From that, the vast majority of the traffic (97.06%) are I-format messages. 90% of those are spontaneous packets (cause of transmission (COT) code '<3> spontaneous'). That means that 88% of the traffic from agents to controllers is generated by the occurrence of a particular event. Thus, most of Power's agents report changed data to the controller rather than sending static data (cyclic/periodic). For example, a change in the state of a binary point (e.g., a switch that passes from off to on), or in the case of analog points, when the values exceed a certain threshold (e.g., a frequency passes the 60.2 Hz threshold). In addition, as shown in Table 2, around 60% of the packets are time-stamp data. This is important for logging events, forensic analysis, and real-time control.

This indicates a network that prioritizes real-time monitoring and rapid response to changes in order to be able to react more quickly to real-time changes, making it an event-driven network. This is crucial to the stability of the power grid.

Minimal Overhead In an event-driven architecture resources are utilized more efficiently, given that data transmission is primarily triggered by significant events. This minimizes the amount of 'noise' in the system by reducing the transmission of redundant or unnecessary data. The approach also ensures that the network bandwidth is optimally used, making it easier to scale the system in the future or allocate bandwidth for other critical applications. Moreover, by focusing on real-time, event-triggered data, the system is better equipped to quickly identify and respond to abnormal conditions, thereby enhancing the overall reliability and security of the power grid.

Only one-fifth of traffic is generated by the controller, mainly for flow control: 80.7% for message control (*S-Format*[3] packets), and connection control (16.7%

[3] *S-format* is a control field packet used for controlling the transport of information (ASDU packets). This protects against loss and duplication of I-format messages.

U-frame[4]). This means that most of the C2a data (80.3%) is dedicated to message acknowledgment, which is a small percentage (16%) of total traffic (a2C + C2a).

By inspecting the IEC 104 header data, we observe that the controller has a large acknowledgment window (w)[5] equal to 8. This means more data packets can be in flight before requiring an acknowledgment, resulting in better throughput. Additionally, an agent can send multiple packets before waiting for an acknowledgment, thus reducing round-trip time and improving latency.

Gas ($a2C < C2a$) In the Gas network, more traffic is sent out from the controller than what is sent by the agents. There is a noticeable difference in the distribution of packets between controller-to-agent ($C2a$) and agent-to-controller ($a2c$): $a2C = 45.8\%$ and $C2a = 54.2\%$. This imbalance can be attributed to two factors.
(1) *Polling Mechanism:* The controller employs Interrogation Commands to solicit data from the agents. These commands are sent as I-Frame packets, increasing the packet count in the $C2a$ direction.
(2) *Acknowledgment Scheme:* Unlike the controller, which acknowledges the receipt of each I-frame from the agent with an S-frame ($w = 1$), the agent does not reciprocate. When the agent receives an I-frame (Interrogation command) from the controller, it sends back the requested data in an I-frame but does not acknowledge it with an S-frame. This unidirectional acknowledgment contributes to the imbalance in packet distribution.

In essence, for each cycle of data exchange initiated by a polling command, the controller sends two types of frames (first an I-frame to request data and then an S-frame to acknowledge receipt) while the agent only sends one I-frame in response. This results in a higher packet count in the $C2a$ direction. We add the following two observations.

One-to-One Acknowledgment Unlike the Power network, which waits until it receives 8 I-frames before it sends back an acknowledgment ($w = 8$), the Gas network operates with a smaller (average) window size of just 1. This is due to the inter-arrival time between I-frames being so large that they need to be acknowledged separately.

A non-standard use of the IEC 104 protocol In the Gas network, the controller utilizes station interrogation (Interrogation commands) instead of Cyclic data transmission to synchronize the process data of the agents. The difference is that Interrogation commands acquire a full set of data, while polling only gets the data that is of interest. Interrogation commands are used to update the controller after initialization or after data loss or corruption of data [8]. On the other hand, cyclic data is used to provide periodic updating of the process data to current values.

[4] *U-format* control field used to control the connection between stations. It is used as a start-stop mechanism for information flow. As a heartbeat to check connection. Also, as a mechanism for changeover between connections without loss of data when there are multiple connections available between stations.

[5] w specifies the maximum number of received I-format APDUs that the receiver should ACK at the latest. e.g., a $w = 8$ means that the controller will send to the agent an S-format message to ACK the last 8 I-format messages it receives.

Interrogation commands are event-based (loss of communication) or manually initiated (start a communication). Another difference between data acquisition by the Interrogation command and cyclic is that the former requires a request, while the latter does not. Interrogation commands are used to poll data from the agent, while cyclic does not require any commands; it is generated automatically by the agent (less traffic). Polling data by using interrogation commands is like a request/response; however, the agent can send the response in several messages, unlike Modbus/TCP, which sends the response in one message. The Gas network does not use cyclic data transmission, only general interrogation for polling data from agents. This is a non-standard use of the IEC 104 protocol in an ICS. It appears to be using a legacy approach like the one used in Modbus/TCP, but it implements it in IEC 104, without taking advantage of the transmission mechanism that the more modern IEC 104 protocol provides.

Water ($a2C = C2a$) In the final case, our Water network has an equal amount of packets being sent by the controller to the agent, as well as from the agent to the controller. Like the Gas network, our Water network operates on a polling mechanism. Given its request-response protocol architecture, Water exhibits an equal traffic flow in both directions ($a2C = C2a$). In essence, for every data report the agent sends, the controller initiates the communication by sending a request. This implies that the controller frequently queries the agent to retrieve the latest state information or execute specific commands.

Response Granularity Both Water and Gas utilize a polling-driven mechanism, but they diverge in how responses are sent by agents. For instance, in Gas, an agent might respond to an Interrogation command with 5 packets, each containing 21 IOA of ASDU type 9. This results in 7 packets for the entire transaction: 3 for an Act, ActCon, ActTerm packet, and 4 for the actual data points. This increases the total number of packets in the transaction for one request. In contrast, Water adheres to a one-to-one request-response model, with each request from the controller receiving a single packet response from the agent. Consequently, a complete transaction in Water consists of just 2 packets: one for the request and one for the response. This streamlined approach minimizes packet loss and reduces network latency, making it more suitable for the time-sensitive operations in a water treatment facility.

5 Discussion

In the previous section, we showed that not all SCADA networks are built the same way. In different domains we observed differences in their topology, the industrial protocols, the traffic characteristics, and the protocol data types used. These results can help us dispel past misconceptions about SCADA networks.

5.1 Dispelling Misconceptions

SCADA communications have been analyzed by the community for some 20 years, but previous research has focused on results from testbeds or only single real-world networks. Therefore, the observations of previous research are

sometimes inaccurate or not representative of diverse, real-world, operational conditions.

The common wisdom we have seen repeated in the literature is that all SCADA networks are similar, and they tend to be painted under the same broad strokes. Below we discuss some observations from past research and compare them with our observations.

Polling "Due to the polling mechanisms typically used to retrieve data from field devices, industrial control network traffic exhibits strong periodic patterns" (Barbosa *et al.* 2012) [5]. "most of the SCADA traffic is expected to be generated periodically due to the polling mechanism used to gather data." (Barbosa *et al.* 2016) [6]. "Due to the use of request-response communication in polling, SCADA traffic exhibits stable and predictable communication patterns.". (Lin *et al.* 2018) [17].
Our Observations Our analysis reveals that Request/Response is not always the mode of communication employed. Specifically, only Modbus/TCP has a flow that reflects a pure polling flow pattern with an equal percentage of traffic in both directions.

In IEC 104 networks, and all networks employing protocols that implement the report-by-exception paradigm, this assumption does not hold.

Flow Direction "the bulk of the traffic is generated from field devices regularly reporting data to the master and the master occasionally sending commands as needed" (Formby *et al.* 2017) [9].
Our Observations We saw in Section 4.5 that for gas, this relationship can be reversed. In this case, the controller (i.e., the IEC 104 master) sends more data to the other endpoints of the connections (C2a $>$ a2C). Furthermore, in the water treatment scenario (Modbus/TCP), the controller (i.e., the Modbus/TCP master) periodically queries sensor readings. While the responses containing the readings are bigger packets, communication is *always* initiated by the controller.

Simplicity of Network Topology "control systems tend to have static *topology, regular traffic, and simple protocols.*"(Cheung *et al.* 2006) [7].
Our Observations The *topology* of SCADA networks is dynamic. Mai *et al.*, 2020 [18] shows topological differences in a bulk power grid over two consecutive years. They found that processes such as energy dispatch can affect the topology daily by adding or removing generation nodes regularly according to the demand needs. In addition, the frequency of maintenance in electrical elements, such as generation machines and transformers, removes nodes temporally. Expansion projects, which add new nodes to an existing network, are not infrequent. For instance, we observe addressing changes and maintenance work in the gas distribution network [21].

Additionally, the IEC 104 protocol is newer and more complex when compared to Modbus/TCP. Features like spontaneous messages (i.e., agents pushing measurements to the control server when the signal changes) allow, among others, for monitoring large(r) networks (including e.g., the wide-area networks of

the power grid and the gas distribution network) without an unmanageable increase in the amount of traffic. The development of IEC 104 (or IEC 101) is the result of restrictions with existing protocols. It is reasonable to assume that new (and potentially proprietary) SCADA protocols are developed to meet new requirements and usage scenarios, e.g., for mobility and automotive scenarios.

"SCADA systems typically use primary-backup approaches to provide disaster recovery capabilities. Specifically, a hot backup of the central control server (the SCADA master) can take over immediately if the primary SCADA master fails, and in many SCADA systems, a cold-backup control center can be activated within a couple of hours if the primary control center fails." (Babay *et al.* 2018) [4].

Our Observations As we showed in this paper, there are several SCADA topologies, and some do not follow the typical primary-backup paradigm. As we saw in the power dataset, the backup controller in a $K_{2,q}$ network is not in hot or cold stand-by configuration; it is a secondary server helping also load-balancing and taking an active part in the monitoring of the system even if the primary server is operational.

More in general, IEC 104 has built-in support for redundancy. The features defined in the IEC 104 protocol soften the definition of a primary control and a (hot) backup server, as both controllers are regarded as equal and can stem the load alone, if necessary. Whether a controller is the primary control server or backup for a specific agent is a matter of choice (i.e., can be selected arbitrarily).

Simplicity of Traffic Patterns "control systems tend to have static *topology, regular traffic, and simple protocols.*"(Cheung *et al.* 2006) [7].

Our Observations Fig 3b shows that in several cases there are no regular traffic patterns. Protocols that use spontaneous transmission, such as IEC 104, present high variability in traffic because the data report depends on the status of the physical process being monitored. Furthermore, some SCADA networks are composed of heterogeneous devices configured by different contractors, as we observed in the power transmission network we studied: therefore, they may have different configurations resulting in diverse traffic patterns.

Simplicity of Network Protocols "control systems tend to have static *topology, regular traffic, and simple protocols.*"(Cheung *et al.* 2006) [7].

Our Observations As discussed in our analysis, early SCADA protocols such as Modbus/TCP were fairly simple. However, newer protocols like IEC 104 (or IEC 101 in this regard) have a more complex structure, with an object model defining a rich set of data types, and less simple communication features like spontaneous (asynchronous) messaging.

Protocols like BACNet (a control protocol commonly used for building automation and management) or OPC-UA, for example, support alerting, push-pull notifications and subscriptions, define a variety of different value and data types, reading and writing of files, and encapsulation of foreign protocol messages. Overall, we argue that, due to evolving requirements and usage context, industrial protocols are becoming more complex than what initially expected.

Timing and Periodicity of Traffic "SCADA systems for the power grid must deliver device status updates and supervisory commands within 100-200ms." (Babay *et al.* 2018) [4].
Our Observations As we saw in our analysis, requirements for data reporting can change significantly, not only among networks but even among different endpoints in the same network. Most of the status updates in our networks took more than 200ms.

While substation equipment can report their status back to a control center with latency $>$ 200ms, protection equipment within the substation network requires faster device status updates (e.g., in the range of 10 ms).

Summarizing, we contend that the prevailing academic perspective on SCADA protocol usage in real systems is analogous to observing just a fraction of a larger puzzle. Often, researchers draw conclusions based on an isolated SCADA network, overlooking the broader context because they don't have access to other operational networks. Our paper aims to shed light on the multifaceted and evolving nature of SCADA systems within the power grid, striving for a more comprehensive understanding.

5.2 Limitations

Our three datasets, even if meaningful in size and diversity, do not represent the entirety of all SCADA networks. While we expect to see similarities in two IEC 104 power transmission networks, our study already showed that networks using IEC 104 can be build differently, i.e., with different design strategies in mind, and thus make use of IEC 104 differently.

Neither the power nor the water dataset span a timeframe that would allow us to make conclusive statements over the entire physical process. Changes in demand throughout the day, and depending on the time of year, would require a much longer capture duration for the power grid. Though, from another paper we know that we can expect that 24 hours are likely enough to contain at least one entire (water treatment) process cycle[10] – though details may depend on other factors including plant location and water purity.

Nevertheless, even if our data might not provide a complete overview of the systems, to date, this is the most comprehensive comparison available. Our comparison offers valuable insights into the diversity of SCADA traffic within Industrial Control Systems. Our dataset is the most diverse ever reported in an academic context, boasting broad coverage across different organizations, device types, and protocols within ICS.

6 Conclusion

In this paper, we uncovered revealing patterns and operational behaviors across different Industrial Control Systems. Through detailed analysis, it became evident that while the power and water treatment networks adhered to conventional protocol applications, the gas distribution network deviates, reflecting intriguing operational choices, particularly in its data synchronization strategy.

Notably, some degree of consistency was observed in the ICS traffic across all networks: approximately a quarter of their traffic exhibited uniform packet sizes. Furthermore, a predilection for small packet sizes, falling within the 0-100 bytes range, was dominant in all three networks, accounting for more than half of their communications. This can be attributed in part to the specification of the two ICS protocols at hand (IEC 104 and Modbus/TCP), defining a relatively small maximum packet size ($\sim$255 bytes). More in general, we argue that parameters such as the average size of packets are influenced more by the ICS protocol and process control technology being used than the specific process being controlled. When exploring transmission timings in ICS, data update frequencies typically spanned from seconds to minutes, with millisecond-order updates being rare exceptions, as exemplified by the relatively inactive gas network.

In general, the update frequency of process data is a parameter that depends on the process control application being observed. In the three examples we observed, real-time fast updates were not necessary nor possible in the case of large geographically distributed networks. In other application contexts (such as the communication among protection equipment within an electrical substation), real-time and fast updates are necessary and implemented by means of specific real-time protocols (such as GOOSE).

IEC 104 networks, a pivotal focus of our study, revealed two consistent operational tendencies: a minimal inter-arrival time that hovers around one second and an (average) maximum packet size capped at 200 bytes. Additionally, our findings highlighted the predictability of packet sizes for polling-oriented networks: here, the controller sends a fixed-length request and receives a fixed-length response, depending on the requested data. In networks with event-driven notifications, the arrival of messages is not easily predictable and message sizes present a richer tapestry of diversity and fluctuation.

This exploration emphasized the heterogeneity within SCADA networks and the importance of customized and specialized approaches for each infrastructure. Our findings challenge generalized views on SCADA networks, advocating for more nuances in the studies of these industrial systems networks.

A better understanding of the nuisances of SCADA networks will give the research community better tools to both evaluate the practical applicability of network and security monitoring approaches and to design new security monitoring approaches based on assumptions that hold in a large(r) amount of real-world setups.

Acknowledgements This work was supported in part by NSF CNS-1929410, CNS-1931573 and by the INTERSECT project, Grant No. NWA.1162.18.301, funded by the Dutch Research Council (NWO). Any opinions, findings, conclusions, or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the funding organizations.

References

1. Iec 60870-5-104 (jun 2006), https://webstore.iec.ch/publication/3746

2. Modbus application protocol specification v1.1b3 (apr 2012), https://www.modbus.org
3. Babay, A., Schultz, J., Tantillo, T., Beckley, S., Jordan, E., Ruddell, K., Jordan, K., Amir, Y.: Deploying intrusion-tolerant scada for the power grid. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 328–335 (2019). https://doi.org/10.1109/DSN.2019.00043
4. Babay, A., Tantillo, T., Aron, T., Platania, M., Amir, Y.: Network-attack-resilient intrusion-tolerant scada for the power grid. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 255–266 (2018). https://doi.org/10.1109/DSN.2018.00036
5. Barbosa, R.R.R., Sadre, R., Pras, A.: Difficulties in modeling scada traffic: A comparative analysis. In: Taft, N., Ricciato, F. (eds.) Passive and Active Measurement. pp. 126–135. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
6. Barbosa, R.R.R., Sadre, R., Pras, A.: Exploiting traffic periodicity in industrial control networks. International Journal of Critical Infrastructure Protection **13**, 52–62 (Jun 2016). https://doi.org/10.1016/j.ijcip.2016.02.004, https://linkinghub.elsevier.com/retrieve/pii/S1874548216300221
7. Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Valdes, A., Skinner, K.: Using model-based intrusion detection for scada networks. Proceeding of the SCADA Security Scientifc Symposium p. 12 (2007)
8. Clarke, G., Reynders, D., Wright, E.: Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems (01 2004)
9. Formby, D., Walid, A., Beyah, R.: A case study in power substation network dynamics. Proc. ACM Meas. Anal. Comput. Syst. **1**(1) (jun 2017). https://doi.org/10.1145/3084456, https://doi.org/10.1145/3084456
10. Hadžiosmanović, D., Sommer, R., Zambon, E., Hartel, P.H.: Through the eye of the plc: Semantic security monitoring for industrial processes. p. 126–135 (2014). https://doi.org/10.1145/2664243.2664277
11. Harsh, V., Jyothi, S.A., Godfrey, P.B.: Spineless data centers. p. 67–73. HotNets '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3422604.3425945, https://doi.org/10.1145/3422604.3425945
12. Hodo, E., Grebeniuk, S., Ruotsalainen, H., Tavolato, P.: Anomaly detection for simulated iec-60870-5-104 trafiic. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ARES '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3098954.3103166, https://doi.org/10.1145/3098954.3103166
13. Hoffman, A.J.: On the Line Graph of the Complete Bipartite Graph. The Annals of Mathematical Statistics **35**(2), 883 – 885 (1964). https://doi.org/10.1214/aoms/1177703593, https://doi.org/10.1214/aoms/1177703593
14. Hoyos, J., Dehus, M., Brown, T.X.: Exploiting the goose protocol: A practical attack on cyber-infrastructure. In: 2012 IEEE Globecom Workshops. pp. 1508–1513 (2012). https://doi.org/10.1109/GLOCOMW.2012.6477809
15. Jung, S.S., Formby, D., Day, C., Beyah, R.: A first look at machine-to-machine power grid network traffic. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm). pp. 884–889 (2014). https://doi.org/10.1109/SmartGridComm.2014.7007760
16. Lin, C.Y., Nadjm-Tehrani, S.: A comparative analysis of emulated and real iec-104 spontaneous traffic in power system networks. In: Abie, H., Ranise, S., Verderame, L., Cambiaso, E., Ugarelli, R., Giunta, G., Praça, I., Battisti, F. (eds.) Cyber-Physical Security for Critical Infrastructures Protection. pp. 207–223. Springer International Publishing, Cham (2021)

17. Lin, C.Y., Nadjm-Tehrani, S., Asplund, M.: Timing-based anomaly detection in scada networks. In: Critical Information Infrastructures Security. pp. 48–59. Springer International Publishing, Cham (2018)
18. Mai, K., Qin, X., Ortiz, N., Molina, J., Cardenas, A.A.: Uncharted networks: A first measurement study of the bulk power system. In: Proceedings of the ACM Internet Measurement Conference. pp. 201–213. ACM, Virtual Event USA (oct 2020). https://doi.org/10.1145/3419394.3423630, https://dl.acm.org/doi/10.1145/3419394.3423630
19. Maynard, P., McLaughlin, K., Haberler, B.: Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks. In: 2nd International Symposium for ICS & SCADA Cyber Security Research 2014. BCS Learning & Development (sep 2014). https://doi.org/10.14236/ewic/ics-csr2014.5, http://ewic.bcs.org/content/ConWebDoc/53228
20. Mehner, S., Schuster, F., Hohlfeld, O.: Lights on power plant control networks. In: Hohlfeld, O., Moura, G., Pelsser, C. (eds.) Passive and Active Measurement. pp. 470–484. Springer International Publishing, Cham (2022)
21. Qin, X., Rosso, M., Cardenas, A.A., Etalle, S., den Hartog, J., Zambon, E.: You Can't Protect What You Don't Understand: Characterizing an Operational Gas SCADA Network. In: 2022 IEEE Security and Privacy Workshops (SPW). pp. 243–250. IEEE, San Francisco, CA, USA (May 2022). https://doi.org/10.1109/SPW54247.2022.9833864, https://ieeexplore.ieee.org/document/9833864/
22. Roig, P.J., Alcaraz, S., Gilly, K., Juiz, C.: Modelling a leaf and spine topology for vm migration in fog computing. In: 2020 24th International Conference Electronics. pp. 1–6 (2020). https://doi.org/10.1109/IEEECONF49502.2020.9141611
23. Wressnegger, C., Kellner, A., Rieck, K.: Zoe: Content-based anomaly detection for industrial control systems. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 127–138 (2018). https://doi.org/10.1109/DSN.2018.00025
24. Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Pranggono, B., Wang, H.F.: Intrusion detection system for iec 60870-5-104 based scada networks. In: 2013 IEEE Power Energy Society General Meeting. pp. 1–5 (2013). https://doi.org/10.1109/PESMG.2013.6672100

A IEC 60870-5-104

IEC 104 is an application layer protocol standardized by IEC 60870-5-104 [1]. Designed for the monitoring and control of industrial systems, it finds widespread application in sectors such as power grids and gas systems. Operating over TCP/IP, it employs a client/server model for communication.

There are distinct features of the IEC104 protocol:

(1) **Message Types**: Supports both synchronous and asynchronous messages, often referred to as spontaneous or periodic messages. (2) **Balance and Unbalance Communications:** In balanced communication, either the controller device or the agent device can initiate the interaction. In contrast, unbalanced communication allows only the controller device to initiate, with the agent responding. (3) **Message Attributes:** IEC104 messages can carry timestamps and quality attributes, enhancing the information's reliability and context. (4) **Synchronous and Asynchronous Modes:** In the synchronous mode, the agent sends a new message after a fixed period. However, in the asynchronous

mode, the agent sends a message whenever a variable's value strays from a predefined deadband.

Inside these TCP packets of IEC 104, there is one or more Application Protocol Data Units (APDUs). Each APDU is composed of: (1) **Application Protocol Control Information (APCI):** This acts as the header of the message and is essential for the proper transmission and receipt of the message. (2) **Application Service Data Unit (ASDU):** This is the main content of the message and carries sensor and control data that is shared between the field agent and the controller.

APDUs are categorized into three types:

- **I-Format APDUs:** These are the primary carriers of sensor and control data. An ASDU within this format includes a Data Unit Identifier and Information Objects. Each Information Object is a representation of a specific device in the field, and each one is linked to a unique address called the Information Object Address (IOA). Apart from this, the ASDU holds the Type Identification, which denotes the specific data format or command type, and the Cause Of Transmission (COT) that outlines the reason behind the message's dispatch.
- **S-Format APDUs:** These are simpler and serve as acknowledgments. They are dispatched after a specified number of I-Format APDUs are successfully received by the other end.
- **U-Format APDUs:** These have a special role in managing the overall connection. They can command the beginning or cessation of I-Format APDUs and also transmit keep-alive requests to ensure the connection's stability. When a new connection is initiated, it is in the "STOPDT" State by default.

IEC 104 was designed with reliability in mind. Typically, a primary connection is established between a controller and an RTU. Alongside this, there is a secondary or backup connection with another controller server. While the primary connection handles the main data transfer and acknowledgments (I-Format and S-Format messages) and occasionally U-Frame, the secondary focuses on periodic keep-alive checks (U TESTFR messages) to test the status of the connection. If the backup server ever sends a communication initiation command, roles are swapped, with the backup server taking the primary role and vice-versa [18].

B Modbus/TCP

Modbus is a widely used industrial protocol that is easy to implement, maintain and has open specifications [2]. It has several versions, such as Modbus RTU for serial communication and Modbus/TCP for TCP/IP communication. This paper focuses on Modbus/TCP, which is a client/server architecture with a simple request/response protocol. The controller (client) is the only one that can initiate communication with the agent (server). The agent never sends a message unless requested by the controller. Modbus has four data types: input register,

holding registers, discrete inputs, and coils. The two "register" types are 16-bit elements commonly used for measurement values, while discrete inputs and coils are one-bit elements used for status values. The message structure of Modbus consists of three parts: a header called Modbus Application Protocol (MBAP), a Function Code that identifies different operations, and a Payload (Data) that carries the content of the message. The format and size of the payload depend on the function code.

The MBAP includes the following components:

- **Transaction Identifier:** A numerical identifier to match request and response messages.
- **Protocol Identifier:** This is set to zero for Modbus/TCP.
- **Length:** The number of bytes in the frame.
- **Unit ID:** This is used in serial communication as the address of the device when multiple agents are connected to a single controller. It is set to zero for Modbus/TCP. This includes the Unit ID, Function Code, and Data.

In contrast to IEC 104, Modbus does not include timestamp or quality attributes in its packets. For instance, IEC 104 utilizes the attribute CP56Time2a as a timestamp in the ASDU process with a long time tag, such as "M_ME_TF_1" (No. 36), to register the time when the measurement was taken; thus, there is no assurance that the information object sent is current or that the data is accurate. Furthermore, Modbus does not have a standard method for data object description. For example, to determine if a register value represents a voltage value between 0 and 220 V.