Uncharted Networks: A First Measurement Study of the Bulk Power System

Kelvin Mai*
The University of Texas at Dallas
Kelvin.Mai@utdallas.edu

Xi Qin*
University of California, Santa Cruz
xqin9@ucsc.edu

Neil Ortiz
University of California, Santa Cruz
nortizsi@ucsc.edu

Jason Molina
Independent
jason100molina@gmail.com

Alvaro A. Cardenas
University of California, Santa Cruz
alvaro.cardenas@ucsc.edu

ABSTRACT

In the last two decades, the communication technologies used for supervision and control of critical infrastructures such as the power grid, have been migrating from serial links to Internet-compatible network protocols. Despite this trend, the research community has not explored or measured the unique characteristics of these industrial systems, and as a result, most of these networks remain unstudied. In this paper we perform the first measurement study of a Supervisory Control And Data Acquisition (SCADA) network in the **bulk** power grid. We develop a new protocol parser that can be used to analyze packets not conforming to standards, find attributes to profile the SCADA network, and identify several outliers which underscore the difficulties in managing a federated network where different devices are under the control of different power companies.

CCS CONCEPTS

• **Networks** → **Cyber-physical networks**; **Application layer protocols**; **Network monitoring**; **Network measurement**; Network reliability; • **Security and privacy** → *Intrusion detection systems*; • **Computing methodologies** → **Cluster analysis**; **Markov decision processes**;

KEYWORDS

SCADA, traffic analysis, IEC 104

ACM Reference Format:
Kelvin Mai, Xi Qin[1], Neil Ortiz, Jason Molina, and Alvaro A. Cardenas. 2020. Uncharted Networks: A First Measurement Study of the Bulk Power System. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3419394.3423630

*These authors contributed equally to this work.

IMC'20, October 27-29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423630

1 INTRODUCTION

In the past two decades, long-distance communications in Supervisory Control and Data Acquisition (SCADA) systems have migrated from dedicated serial links, to Internet-compatible networks. Therefore, SCADA communication standards for serial links (such as Modbus, and IEC 60870-5-101) have been updated to support TCP/IP networks (e.g., Modbus/TCP and IEC 60870-5-104).

While modern SCADA systems use Internet-compatible protocols, the network measurement research community has largely ignored these networks. One of the reasons is that companies that manage critical infrastructures, such as power grids, are very conservative in allowing outsiders to gain access to their internal networks. As a consequence, most of the published research related to SCADA networks has relied on simulations and testbeds [14, 17, 20, 23, 25, 26], and are not based on operational systems. The few studies of a real-world power grid consider only the small distribution system [10]; as far as we are aware, there is no published measurement study of the networks used in the core part of the power grid, the so called *bulk power system.* These networks are interesting not only because they operate the most critical component of the power grid, but also because they consist of multiple power companies (with different administrative domains) talking to each other's equipment (in contrast, most other SCADA networks operate within a single administrative domain [5, 10, 19]).

Our contributions include,

- As far as we are aware, we are the first to measure and characterize the SCADA network of the **bulk** power system.
- In particular, we study the IEC 104 SCADA protocol for Automatic Generation Control (AGC) in the bulk power grid. IEC 104 is one of the SCADA protocols attacked during the Ukraine power outages in 2016 [13].
- We study the network with three different approaches: (1) traffic analysis of TCP flows, bandwidth used, and timing characteristics of the packets, (2) analysis of the types of IEC 104 messages exchanged with the help of Markov networks, and (3) analysis of the physical measurements and control commands sent between substations and the control center.
- While overall SCADA networks are more stable and predictable than general computer networks, we still find that because of the federated nature of the network we study (devices under different administrative domains), our network has more interesting behaviors than other SCADA networks.

- We also find evidence of the challenges to upgrade legacy protocols to new standards. In particular we find some non-compliant communications based on IEC 104, and upon further inspection, we identify that these non-standard packets are an attempt to support legacy protocols over TCP/IP.
- In order to understand these non-compliant packets we developed a new IEC 104 parser that we made available to the research community [21].

We believe these observations are an important first step to understand industrial networks and their unique characteristics.

The remainder of the paper is organized as follows: Section 2 gives a general overview of the Bulk Power System. Section 3 details related works. Section 4 describes IEC 104. Section 5 summarizes of our datasets and the network we study. Section 6 focuses on our traffic analysis. Finally, Section 7 concludes our work.

2 POWER SYSTEMS BACKGROUND

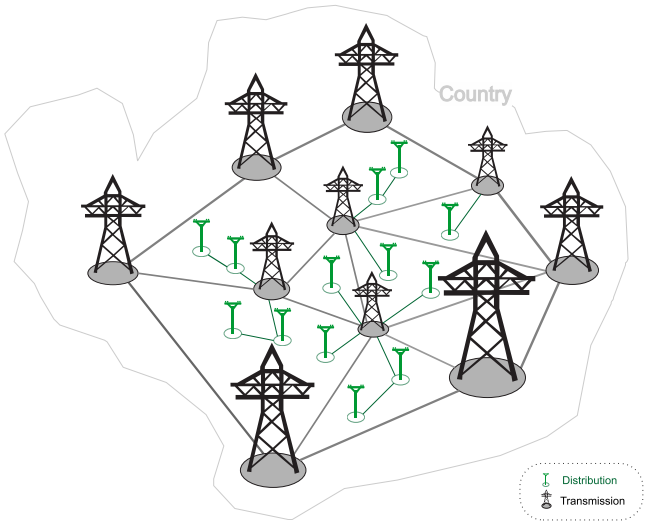

Figure 1: Illustration of the differences between the transmission and distribution systems in a country. While the transmission system (i.e., the bulk power system) is a redundant network covering a large geographical area, distribution systems are independent radial networks covering small geographical areas.

The power grid has three major parts: (1) generation, (2) transmission, and (3) distribution. Electric power is generated wherever it is convenient and economical, and then it is transmitted at high voltages (100kV-500kV) through the transmission network. The transmission system is an interconnected, redundant network that spans large regions (usually one country). Large generation plants and the transmission network (the first two parts of the power grid) are usually referred to as the **Bulk Power System**, and this bulk power system is responsible for the reliable delivery of electricity to large areas. A disruption in the bulk power grid can cause a country-level blackout lasting for several days. In contrast, distribution systems are much smaller, their networks are radial (non-redundant), and a failure in their system usually causes only a localized outage (e.g., a blackout in a neighborhood) lasting only a couple of hours. Fig. 1 and Table 1 illustrate the differences in scale between transmission and distribution networks.

Table 1

	Transmission	**Distribution**
Power [W]	10^9	10^6
Area [km^2]	> 4.67 million	> 10600
Voltage level [kV]	> 110	< 34.5

The bulk power grid is operated by several companies, some of them are electricity generators while others operate a subsection of the transmission system. Each of these companies has their own SCADA system to monitor and control the part of the bulk power grid they are responsible for. Orchestrating the operation of all of these power companies is an entity called **system operator**. In Europe, these operators are called Transmission System Operators (TSO) and there is usually one TSO per country. In the U.S. system operators are called either Regional Transmission Operators (RTO) or Independent System Operators (ISO) depending on whether they administer the power grid among several states (RTO) or if they operate the grid in one state (ISO). For example, the California Independent System Operator (CAISO) operates the power grid for all of the state of California.

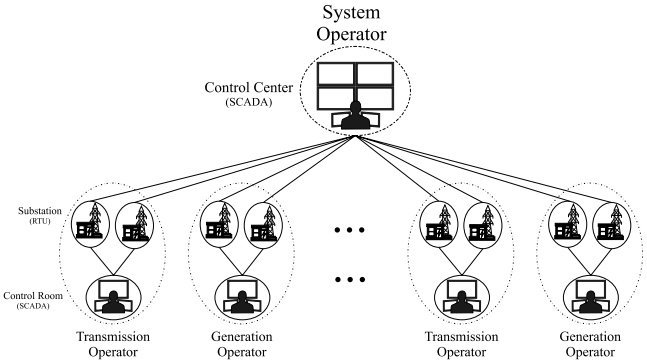

Figure 2: A system operator has to interface with substations controlled by different transmission systems and generators, and as a result the network behavior is more diverse than previously considered.

One of the essential tasks of the bulk system operator is to coordinate the power balance across multiple geographical regions and to maintain the frequency of the system at the desired set point (e.g., 60Hz in the U.S.). To achieve this, they use an algorithm called **Automatic Generation Control (AGC)** which asks different electric generation companies to ramp up or slow down their electricity generation to maintain an adequate power flow balance in the system, and thus satisfy the reliability and market efficiency of the electric power system. AGC uses as primary inputs the frequency of the power grid and the power flow at different power exchange lines.

Because these bulk system operators have to collect sensor data and send control commands to various other companies operating the power grid, they form *federated* SCADA networks, where devices in the network are owned and maintained different administrative authorities. Fig. 2 illustrates this setting, where at the bottom of the figure we can see the SCADA system of each of the *local* operators (transmission companies, and large generation companies) and at the top we see how the SCADA system of the *system* operator connects to the substations of the other companies. The top SCADA system is the one we study in this paper.

3 RELATED WORK

Previous measurement studies of computer networks related to the power grid fall into three categories: (1) use of emulated/simulated networks (i.e., confined to a laboratory environment, or a testbed); (2) insufficient details of the system and network; (3) study a relatively small part of an operational power grid system, i.e., distribution networks.

Analysing emulated or simulated data is the most popular approach, as researchers can configure their equipment however they want. This line of work includes a testbed at KTH [17, 25] simulated IEC 104 networks [14] or emulated IEC 104 networks through Qtester [20].

Some papers study operational power grids, but they do not give details of the system under study. For example, Yang et al. [29] capture network traffic data from a real-world IEC 104 system without adding details of the system they are analyzing. Similarly Wressnegger et al. [28] indicate that their network capture comes from a power plant, but they do not specify which network protocol is used or add any details of the network.

Perhaps the work most closely related to ours is Formby et al. [9, 10] and Irvene et al. [15], where they analyze a real-world electric power distribution substation that uses the DNP3 industrial control protocol. Both of these works study the same distribution system, which is a relatively small component of the power grid.

In contrast, our dataset is captured from a **Bulk Power System**, which as we discussed before, is the core component of large-scale power systems. In addition, our dataset includes data used for Automatic Generation Control (AGC) and focuses on the IEC 104 SCADA protocol, which recently gained more visibility as the target of the attacks in Ukraine [13].

More importantly, our dataset includes network traffic not only from a single power operator but rather, from a regional power balancing authority. The balancing authority coordinates multiple power operators over a broad geographical area serving a population of about 40 million people. In addition, our data captures were obtained in two different years, giving us the unique ability to compare the changes and similarities of the network over longer periods of time when compared to previous work. In particular, this paper extends our preliminary results [18] by providing an in-depth look at the network characteristics and dynamics of bulk power systems controlling power generators in a wide area network.

4 IEC 104

IEC 60870-5-101 (IEC 101) [6] was originally developed by the International Electrotechnical Commission (IEC) in 1995 and was amended in 2000 and 2001 to provide a standard that enables basic telecontrol messages between **control stations** (e.g., SCADA centers) and **outstations** (e.g., devices in substations such as **Remote Terminal Units (RTU)**) via a permanently connected communication link over the telephone network i.e., modem circuit. With the prevalence of TCP/IP networks, it became apparent that SCADA systems needed to adapt their protocols to these networks. Therefore in 2000, IEC 60870-5-104 (IEC 104) was introduced as a way to transport IEC 101 telecontrol messages over TCP/IP using port 2404. IEC 104 encapsulates modified IEC 101 telecontrol messages into a TCP packet.

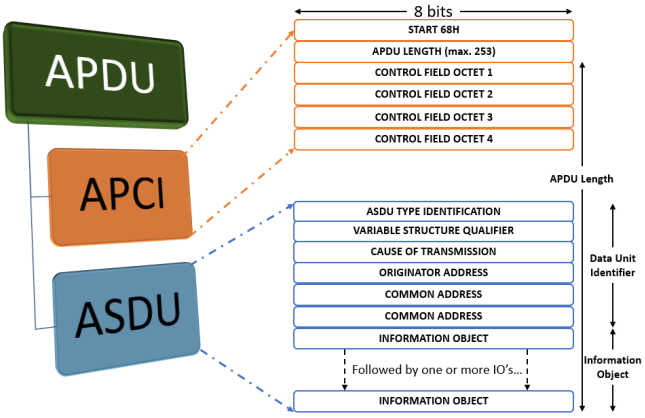

Figure 3: IEC 104 APDU Octets Structure

The TCP payload of an IEC 104 packet contains one or more **Application Protocol Data Units (APDUs)**. The first part of the APDU is called the **Application Protocol Control Information (APCI)**, which acts as the header of the message, and the second part is called the **Application Service Data Unit (ASDU)**–this second part carries the sensor values and control messages between RTUs and control servers. APCI and ASDU fields are shown in Fig. 3.

There are three types of APDUs:

I-Format APDUs are used to carry sensor and control data between endpoints. ASDUs are composed by a Data Unit Identifier (DUI) and by Information Objects (IO) as illustrated in Fig. 3. Each IO represents a specific device in the field which is associated to a unique address called **Information Object Address (IOA)**. The first ASDU octet is **Type Identification** (typeID) which defines the exact data format or command type that follows. For example, *"Measured value, short floating point number"*, or *"Set point command, scaled value"*. IEC 101 defines 127 TypeIDs from which IEC 104 only supports 54. In addition, an ASDU also contains the **Cause Of Transmission (COT)**, such as periodic (e.g., a periodic reporting of the voltage), spontaneous (e.g., the current exceeded a pre-configured threshold), or interrogation (exchanging values based on a request by the other party). In short, ASDU typeID specifies **"what"** type of data/command is being sent and COT specifies **"why"** it is being sent.

S-Format APDUs are basically acknowledgments after a specific (but configurable) number of I-Format APDUs have been received.

U-Format APDUs provide three connection control functions: (1) they can start the transmission of I-Format APDUs via a `STARTDT act` message (which is acknowledged with a `STARTDT con` message); (2) stop the transfer of I-Format APUDs with the `STOPDT act` message (also acknowledged with a `STOPDT con` message), and (3) send keep-alive connection requests with the `TESTFR act` message (which is acknowledged with a `TESTFR con` message). Newly established (or switchover) connections are by default in a `"STOPDT"` state.

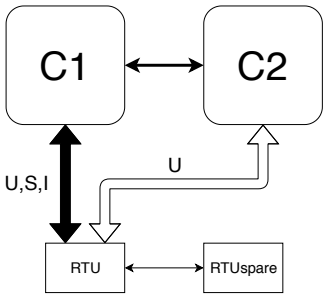

Figure 4: Primary and Secondary Connections in IEC 104.

In high reliability networks, IEC 104 typically maintains a primary connection between a server and an RTU and a secondary (redundant connection) with another server as illustrated in Fig. 4. The primary connection is used to send I messages, which also include S messages (acks) and occasionally U messages. The secondary connection only sends periodic U `TESTFR` messages to test the status of the connection (a keep-alive message). If at any point in time the backup control server C2 sends the U `STARTDT con` message, then the connection to server C2 becomes the primary connection or the RTU and the connection to C1 becomes the secondary connection. This behavior and the establishment of a TCP flow is determined by four timers.

- T_0: Timeout of connection establishment (default at 30 sec). Expiration of this timer will trigger a TCP-SYN request to establish a new connection.
- T_1: Timeout of send or test APDUs (default at 15 sec). Expiration of this timer will trigger an active close request, or a connection change request by a controlling station, resulting in the start of a new redundant connection and an automatic switch over.
- T_2: Time out for acknowledgements (default at 10 sec and T2 < T1). Expiration of this timer will cause the receiver to send an S-Format APDU.
- T_3: Time out (default at 20 sec) for sending keep-alive messages in a connection that is not currently sending any format(I/S/U) of APDUs.

5 DATASET AND NETWORK DESCRIPTION

In the power system we study, there are three types of substations: (1) those with serial links (IEC 101), (2) those with IEC 104, and (3) those who do not connect directly to the system operator, they only connect to the power company who owns the substation (the system operator eventually gets access to these substations by talking with the control center of the power company via the ICCP industrial protocol). The substations that use IEC 101 or IEC 104 to connect directly with the system operator, do so because to allow AGC control of their generators. In this paper we focus on these substations; in particular the ones with IEC 104 (we cannot observe the substations that use IEC 101 because they do not show up in our network tap).

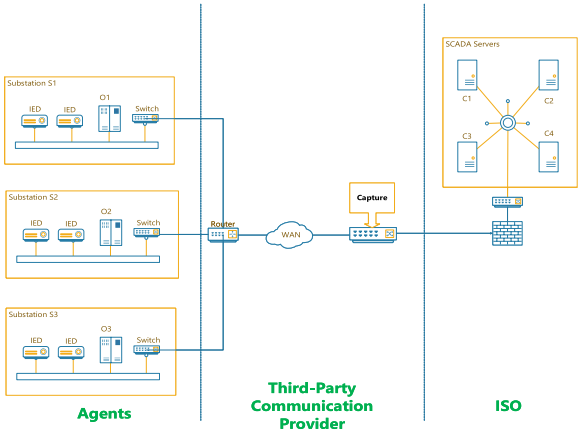

Figure 5: Tap between substations and SCADA server.

Fig. 5 shows our network tap within the SCADA network. On the left side we have substations, where information about the power grid is captured by Intelligent Electronic Devices (IEDs) and then sent to RTUs; we label RTUs with an Ox where x is a number identifying each RTU/Outstation (RTUs are called Outstations in the IEC 104 standard and we will use the latter term in the rest of the paper). The Outstations then send information to the SCADA servers through a private IP network using the IEC 104 standard. In addition to IEC 104 traffic, our capture included other industrial protocols over TCP/IP such as ICCP (communications between SCADA servers of different companies) and C37.118 (phasor measurement units reporting data to the SCADA server). We leave the analysis of these other protocols for future studies. Our datasets were collected in two different years with a one year gap. Overall we made 5 captures in different days during the first year totaling approximately 8 hours, and 3 captures in different days of the second year totaling approximately 3 hours. In the rest of this paper, will refer to datasets collected in the first year as "Y1"and the second year as "Y2".

Before we discuss our analysis, we start with a set of hypothesis and questions about our data to guide us in our measurements.

Hypothesis 1: Among the published research in SCADA security, the consensus is that SCADA networks are fairly stable and predictable [3, 27], given their machine to machine communication, and the long-term investments in equipment in the power grid [1] (10-50 years), when compared to regular information technology networks. In this paper we study this assumption and discuss the changes of the network over the years and the things that remain the same.

Hypothesis 2: One of the reasons for standard-based communications is to have a unique and reliably way for accessing

different equipment from different vendors. We expect end point devices that communicate using IEC 104 to have communications readable by IEC 104 compliant devices.

Hypothesis 3: Previous work has shown that SCADA networks tend to have long TCP/IP flows, where a TCP connection is established and it is kept alive for days or even weeks of communications between field devices and SCADA servers. We expect our SCADA network will also have long TCP flows.

Hypothesis 4: Given the limited types of messages being exchanged in SCADA systems, we expect we can characterize the types of connections in clear clusters that can give us insights into the operation of the network (traffic analysis, and sequence of messages via Markov models).

Hypothesis 5: SCADA systems monitor the physical world; however, network measurement studies rarely study what we can learn from this. Our hypothesis is that by performing DPI we can learn new "physical" behavior of the underlying power system to help us profile better these systems.

6 NETWORK MEASUREMENT

Fig. 6 shows the network we observed in our datasets from Year 1 (Y1) and Year 2 (Y2). We can see that the control room of the system operator has 4 control servers: C1, C2, C3, and C4. We also observed a total of 27 substations (identified in the Figure as S1-S27).

Most substations are next to a power generator (identified as ovals) and some substations only deal with transmission equipment (identified as semi-circles). This makes sense as the role of IEC 104 for this particular operator, is to monitor and control generators (via AGC). The few substations that do not have generators provide auxiliary network measurements of the bulk power system. Each substation has one or more RTU, and because RTUs are called Outstations in the IEC 104 standard, we identify them in the figure as O1-O58. We can see that each pair of servers (C1/C2 and C3/C4) maintains a primary and a secondary connection to each outstation (as expected by Fig. 4). Finally, each RTU collects measurements from a variety of field devices, from sensors in generators, to circuit breaker information, frequency sensors, etc. These devices are enumerated in the "cloud" attached to each Outstation (RTU).

In order to test our first hypothesis, we first look at the changes of the network over a year. Fig. 6 illustrates several changes from Y1 to Y2. We can see in red the substations and outstations removed from Y1, and in green, the new outstations that had been added to Y2. The arrows associated with each "cloud" indicate changes in field device measurements that we observed between Y1 and Y2. An upward arrow indicates that we observed more IOAs in Y2 than in Y1, and an downward arrow indicates we observed less IOAs in Y2 (the number of IOAs observed in Y1 are in red and the number of IOAs seen in Y2 are in green).

We asked the bulk system operator about these changes and their answers are summarized in Table 2. There are four different reasons for having new outstations in Y2. The first reason is that there are new substations that came online in Y2. In particular O50, which is associated with substation S24, and O53, associated with substation S27. The power grid operator told us that adding new substations over the years is not uncommon, and in fact this trend is accelerating with the addition of renewable energy. The second reason for additions is that substations with serial links (IEC 101) were updated to TCP/IP networking with IEC 104; these correspond to O52/S23 and O55/S26. The third addition occurred because O54/S25 was undergoing maintenance during the first year of the capture, and that is why we did not see it in Y1. The final reason for additions is a simple one, many substations have backup outstations that can talk to the control servers. In the first year we captured a different set of outstations communicating with the servers, but in the second year we captured their alternate outstation; these include O51, O56, O57, and O58, and similarly some of the removed outstations like O28 were replaced by these redundant RTUs, while others such as O15 have a backup outstation (O9 in this case) which still represents the substation to the control servers. Perhaps the most surprising finding was the removal of O2/S2; the operator told us that this substation had lost their connection and therefore was not monitored by the system operator, but this does not mean the substations was completely unsupervised, as it still presumably has the main connection to the SCADA server of the power company managing the substation. Another reason S2 was not essential for the operator is because it is not a generation substation (i.e., it does not have a generator that can be controlled by their system) and therefore it is one of the auxiliary substations that send data complementing their view of the grid, but the missing data from S2 did not represent a critical component for the operation of the AGC algorithm.

Table 2: Outstations added or removed between Y1 and Y2.

Outstation	Added/Remove	Description
O50, O53	Added	New substations
O52, O55	Added	Updated from 101 to 104
O51, O56, O57, O58	Added	Backup RTU
O54	Added	Under Maintenance in year 1
O15, O20, O22, O28, O33, O38	Removed	Redundant RTU in operation
O2	Removed	Substation without supervision

Overall, we see that 7 substations out of 27 (26%), and more precisely, 14 outstations out of 58 (25%) remained connected and reporting the same number of IOAs in a year. So the answer of whether Hypothesis 1 is validated in this network is not clear; on one side, most of the network changed between two years; however, we can see that the server configuration remains stable, and over 1 out of 4 of the devices in the field remains stable.

6.1 IEC 104 compliance

After identifying the main topological changes between the years, we start parsing the IEC 104 network packets to understand in more detail the behavior of the network. When we attempted to use existing parsers for IEC 104 (e.g., Wireshark) we found that several of the packets in our networks were identified as malformed packets. In particular, outstations O37, O53, O58, and O28 had 100% invalid packets in all our traces; we got two errors (1) invalid IOA addresses and (2) the measurements in I-Format APDUs appeared completely random (rather than stable measurements like voltages or frequencies). We reported our findings to the system operator, but they told us that their SCADA system was receiving valid data from these outstations, so we decided to implement our own parser to inspect these malformed packets in detail. Our parser uses

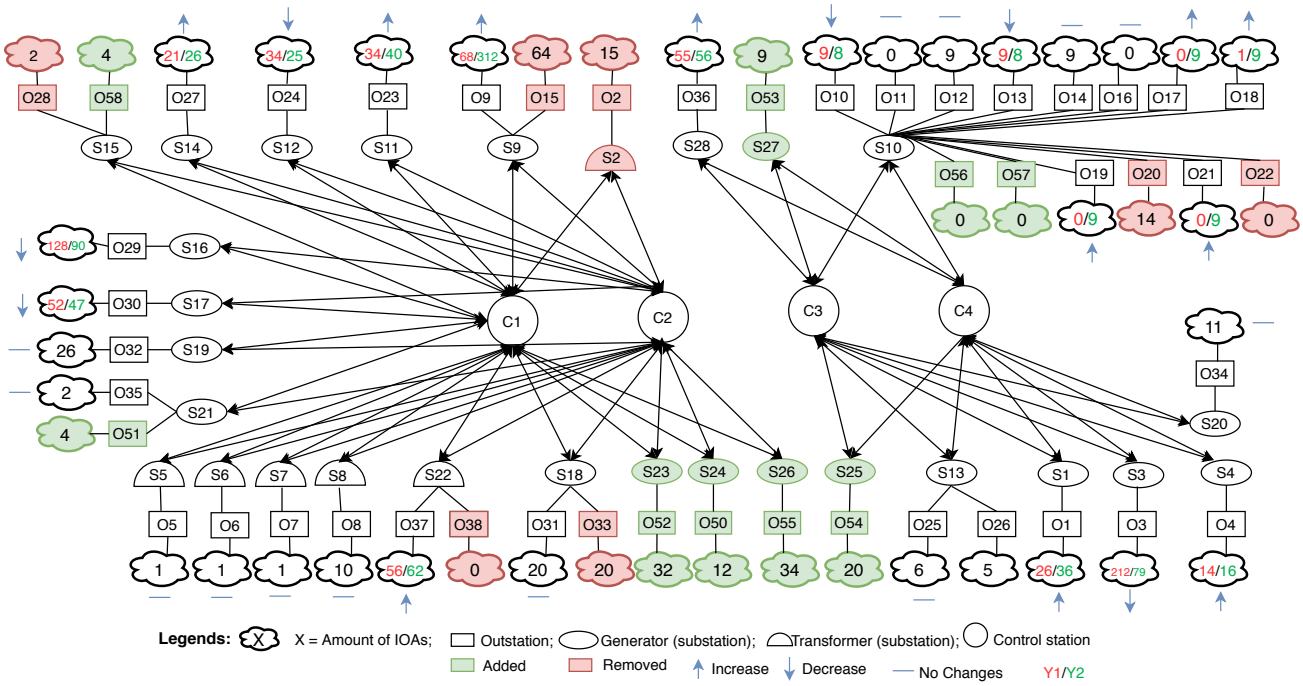

Figure 6: IEC 104 Network Topology: Outstations are represented with 'O'+prefix. Substations as 'S'+prefix, and 'C'+prefix represents control servers. Y1 vs Y2 changes: Substations and outstations removed (red) and added (green). Increase/decrease amount of IOAs (arrow up/down). Clouds indicates total number of IOAs in each outstation.

SCAPY [4], and while SCAPY has a module for IEC104 that follows the standard, we decided to create our own module to analyze those packets that don't follow the standard. The source code of this work is available at our GitHub repository [21].

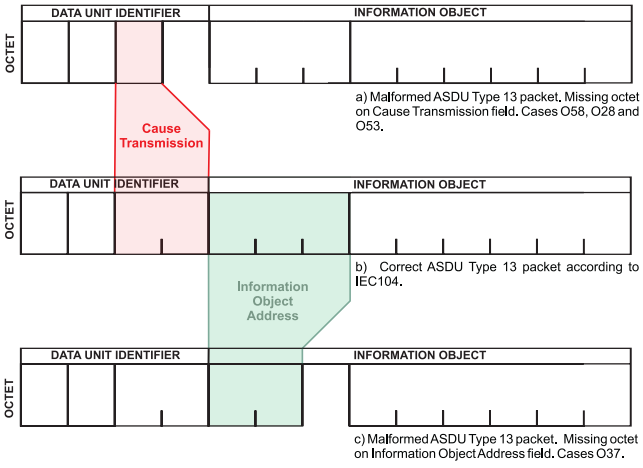

Figure 7: Comparison between a correct IEC104 packet (b) and a malformed packet in Cause of Transmission field (a) and Information Object Address field (c).

We found two reasons these packets did not conform to the IEC 104 standard. First, outstation O37 used an IOA length of just two octets (instead of the standard three octets length for an IOA address). The second set of malformed packets came from outstations O53, O58, and O28, which used just one octet for the "cause of transmission" field, while the IEC 104 standard specifies that the cause of transmission field should be two octets. These two differences are illustrated in Fig. 7.

After spending some time trying to find the reason for these non-standard fields, we found a culprit: the legacy serial protocol IEC 101 allows a single octet as a cause of transmission, and two octets as IOAs. We believe that when the substations were upgraded from IEC 101 to IEC 104, the original configuration of the substation did not change, and therefore they ended up sending IEC 104 packets with IEC 101 legacy options. It appears that the SCADA software vendor is aware of these non-standard options (they might be prevalent in IEC 104 networks) and allows operators to configure the system so that it can read these malformed packets.

Our analysis shows the importance of obtaining data from real-world systems instead of testbeds, and is in direct contradiction with **Hypothesis 2**. In short, Industrial systems that have been in operation for decades and who have migrated from previous technologies, may still retain certain non-standard legacy characteristics that need to be identified in order to properly characterize the network.

6.2 TCP flow analysis

Following up on **Hypothesis 3**, our expectation was to find long-lived TCP flows (defined by the 4-tuple `<srcIP, srcPort, dstIP, dstPort>`). To our surprise, when we attempted to measure the

Table 3: Comparison of the number of TCP short-lived flows and long-lived flows in two years

Year	Y1	Y2
Count of Less-than-one-second Short-lived Flows(proportion)	31614(99.8%)	7937(93.5%)
Count of Longer-than-one-second Short-lived Flows(proportion)	63(0.2%)	549(6.5%)
Count of Short-lived Flows (proportion)	31677 (74.4%)	8486 (93.8%)
Count of Long-lived Flows (proportion)	10898 (25.6%)	560 (6.2%)

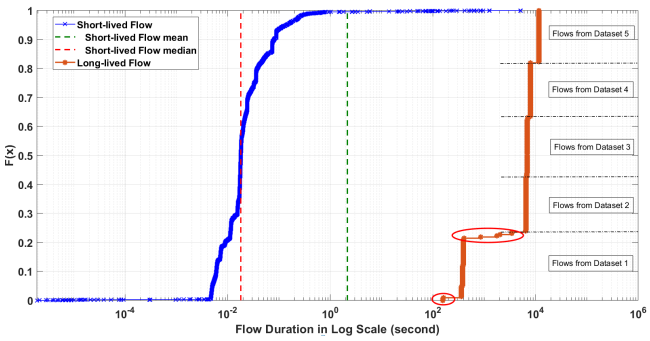

Figure 8: Y1 TCP short-lived flow duration analysis

duration of TCP flows, we found out that 99.8% of the TCP flows in Y1 and 93.5% of the flows in Y2 lasted less than one second. We summarize our findings in Table 3.

This was very counterintuitive for us, so we further divided TCP flows into those where we found a matching SYN and RST/FIN pair in the capture (we call these short-lived flows), and those that either started before our capture or ended after our capture (we call these long-lived flows). The summary of these flows can be seen in the third and fourth rows of Table 3. Fig. 8 shows the duration (seconds in logscale) of the TCP flows. We can see how several of them have very short duration.

Trying to explain the reason for these short lived flows, we found that a small subset of outstations reject backup TCP connections from the server with FIN or RST packets. This behavior is illustrated in Fig. 9. For comparison, the "normal" behavior should be the one in Fig. 4, where the backup server establishes a second connection with the outstation and exchanges U (keep alive) messages to keep the backup connection active.

When we told the power company about these abnormal communications they were not aware of them. For them, the SCADA system still works seamlessly. While the RTU might reject TCP connection attempts for backup IEC 104 channels, when the main connection is teared down, they readily accept the backup connection to the other control server to send I messages. So functionally the system still works as expected from the point of view of the SCADA system.

We then asked if they could reconfigure the misbehaving outstations so that they would accept this backup connection, but their answer that was that they do not own or manage the configuration

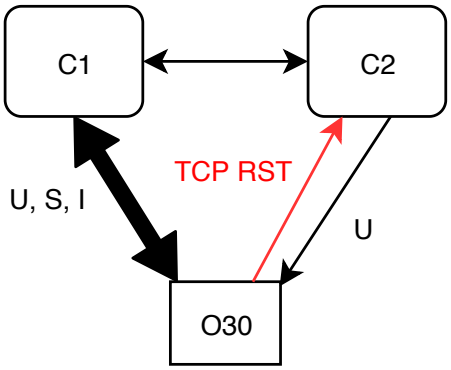

Figure 9: Outlier behavior in clustering analysis.

of the RTUs/outstations. Furthermore we found out that they do not have regulatory power to demand these changes; as long as the application-level behavior is satisfied, all parties are complying with the reliability standards, even if the network behavior underneath has problems. Moreover, we also found evidence that this problem is not only happening at the power system operator we studied. In our analysis of SCADA systems, we got access to the network alerts for IEC 104 networks from Forecout's Silent Defense system [8], and found that one of their hard-coded alerts is this regular connection resets from outstations to servers. This connection reset problem appears to be pervasive in IEC 104 networks, but we do not know the cause for this.

6.3 Traffic Analysis

We now turn to our attention to Hypothesis 4, and look at the types of patterns and profiles that can give us insights into the operation of the network.

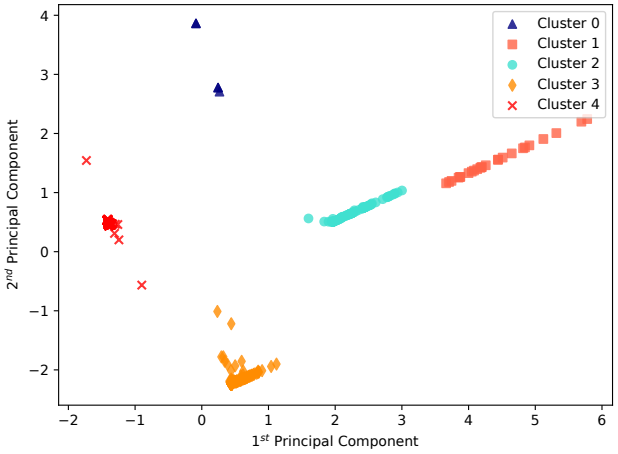

Figure 10: PCA of clustered IEC 104 sessions in Year 1

We define as a *session*, all the packets that are sent in one direction between the same end points. Originally we considered in total of 10 statistical features to investigate, including the transmission direction (is the message coming from the control center or

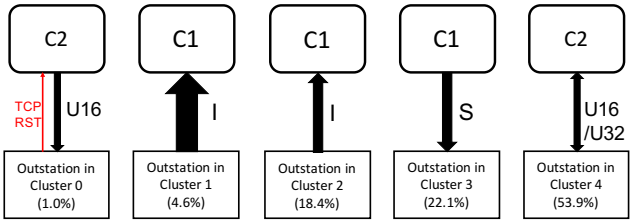

Figure 11: Communication patterns between outstations and control stations in each cluster

from the outstations?), average inter-arrival times, total bytes, total number of packets, and even some features that looked into the APDU information such as the count of IOAs or the distribution of APDUs by type (U/S/I). Using the Silhouette score for each individual feature[22], we pick the features that generate relatively high Silhouette scores and reduce the feature space dimensionality from ten to the following five features:

- Δt_i: average inter-arrival time between two consecutive packets
- num_i: total number of packets sent in the same direction by two end points.
- $percentage_I$: the percentage of I-format data units.
- $percentage_S$: the percentage of S-format data units.
- $percentage_U$: the percentage of U-format data units.

We use K-means++ clustering [16] on these features. To select the number of clusters K we use the Elbow method on the sum of squared error [24], the explained variance[12], and Silhouette scores[22]. These methods suggest that a good number of clusters is K=5. In addition, we use Principle Component Analysis (PCA) [11] to project and visualize our results to a the lower dimension (2D plane). Our clustering results can be seen in Fig. 10.

Inspecting the characteristics of each cluster, we find the following five representative behaviors: (1) Cluster 0 represents (extremely) long inter-arrival arrival times between packets; (2) Cluster 1 contains the largest amount of I-format packets, characterized also by being spontaneous transmissions (as opposed to periodic), (3) Cluster 2 represents the "average" case representing most outstations sending a regular amount of I-format packets, (4) Cluster 3 captures all the acknowledgements (S-format packets) sent from control servers to outstations, and (5) Cluster 4 represents the keep alive messages of the backup IEC 104 connection. Figure 11 summarizes these clusters and their percentages.

We now study in more detail cluster 0, which is an outlier. Cluster 0 contains just two connections characterized by their long inter-arrival times between two consecutive packets (they have the largest Δt_i in our datasets). The sessions in cluster 0 are the packets control server C2 sends to Outstation O30 and the traffic (back and forth) between C4 and O22 in Y1.

Overall, we found that the outlier connection C4, O22 was because of testing procedures (these end points only exchanged four packets in our capture, and the operator confirmed that the RTU was not operational, but being tested), however, the connection C2, O30 is a clear outlier: this connection is a secondary connection as described in Fig. 9, so this abnormal behavior did not affect the operation of the system, however, the interval between U messages was 430s, an interval an order of magnitude higher than the rest of the secondary connections, which had a 30s average time between U messages. We believe this is a misconfiguration of the T_3 timer.

6.3.1 Deep Packet Inspection: Message Sequences. We now focus on deep-packet inspection tools to look in more detail at the nature of communications in the network. Our first goal is to understand what types of sequences are being exchanged between different end points, and in particular, to find if there are sequences of APDUs that can succinctly summarize an end-to-end communication between every pair of devices in SCADA networks.

To model the sequence of APDUs observed in the network, we utilized N-gram models, which were originally proposed for statistical analysis of natural language. Formally, given a finite set Σ, a given language $L(\Sigma)$ is composed of sequences of alphabets over Σ such that $L(\Sigma) \subseteq \Sigma^*$. Let a sequence of words $W = \{w_1 ... w_n\}$ of 'n' words, then the Language Model (LM) probability of this entire word sequence, using chain rule, is:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \quad (1)$$

To create a language model of our dataset, we tokenize each IEC 104 APDU with the elements in Table 4. For example, an S-format APDU followed by an I-format APDU of typeID 36 (i.e., an I-format APDU carrying a measured value in floating point format with time tag) will be represented as a bigram (S, I_{36}), and an I-format APDU of typeID 13 (i.e., an I-format APDU carrying a measured value in a short floating point format without a time tag) immediately followed by another I-format APDU of typeID 13 will be represented as (I_{13}, I_{13}). To compute probability of a given bigram, we use maximum likelihood estimation (MLE), let t be a token, then:

$$P(t_n|t_{n-1}) = C(t_{n-1}t_n)/C(t_{n-1}) \quad (2)$$

Table 4: APDU Token Description

Token	APDU	Description
S	S	Ack of I APDUs
U1	STARTDT act	Start sending I APDUs
U2	STARTDT con	Ack of STARTDT
U4	STOPDT act	Stop sending I APDUs
U8	STOPDT con	Ack of STOPDT
U16	TESTFR act	Test status of connection
U32	TESTFR con	Ack of TESTFR
I_{code} (for code={1,3,5,...,127})*	Variable type	Sensor and Control Values

* A description of all Type IDs can be found in Table 5.

We start our analysis by creating Markov-chain models of message sequences, where each node represents a unique APDU token and tokens are connected by transitional probabilities. Fig. 12 shows two of the simplest expected communication patterns observed in our datasets. The figure on the left shows the expected pattern of a primary connection, where the outstation sends I_{36} APDUs which are periodically acknowledged via S-format APDUs. The image on the right represents the ideal behavior of a secondary (redundant) connection, where the connection keep-alive APDU pairs U_{16} and U_{32} were all that observed repeatedly between control server and outstation. Also seen from the right image is a very low probability of sending repeated U_{16} or U_{32} APDUs (which is an anomaly), but

Table 5: ASDU I-Format Type Identification Codes

Type ID Code	Acronym	Description
1	M_SP_NA_1	Single-point information
3	M_DP_NA_1	Double-point information
5	M_ST_NA_1	Step position information
7	M_BO_NA_1	Bitstring of 32 bits
9	M_ME_NA_1	Measured value, normalized value
11	M_ME_NB_1	Measured value, scaled value
13	M_ME_NC_1	Measured value, short floating point number
15	M_IT_NA_1	Integrated totals
20	M_PS_NA_1	Packed single-point information with status change detection
21	M_ME_ND_1	Measured value, normalized value without quality descriptor
30	M_SP_TB_1	Single-point information with time tag CP56Time2a
31	M_DP_TB_1	Double-point information with time tag CP56Time2a
32	M_ST_TB_1	Step position information with time tag CP56Time2a
33	M_BO_TB_1	Bitstring of 32 bit with time tag CP56Time2a
34	M_ME_TD_1	Measured value, normalized value with time tag CP56Time2a
35	M_ME_TE_1	Measured value, scaled value with time tag CP56Time2a
36	M_ME_TF_1	Measured value, short floating point number with time tag CP56Time2a
37	M_IT_TB_1	Integrated totals with time tag CP56Time2a
38	M_EP_TD_1	Event of protection equipment with time tag CP56Time2a
39	M_EP_TE_1	Packed start events of protection equipment with time tag CP56Time2a
40	M_EP_TF_1	Packed output circuit information of protection equipment with time tag CP56Time2a
45	C_SC_NA_1	Single command
46	C_DC_NA_1	Double command
47	C_RC_NA_1	Regulating step command
48	C_SE_NA_1	Set point command, normalized value
49	C_SE_NB_1	Set point command, scaled value
50	C_SE_NC_1	Set point command, short floating point number
51	C_BO_NA_1	Bitstring of 32 bits
58	C_SC_TA_1	Single command with time tag CP56Time2a
59	C_DC_TA_1	Double command with time tag CP56Time2a
60	C_RC_TA_1	Regulating step command with time tag CP56Time2a
61	C_SE_TA_1	Set point command, normalized value with time tag CP56Time2a
62	C_SE_TB_1	Set point command, scaled value with time tag CP56Time2a
63	C_SE_TC_1	Set point command, short floating-point number with time tag CP56Time2a
64	C_BO_TA_1	Bitstring of 32 bits with time tag CP56Time2a
70	M_EI_NA_1	End of initialization
100	C_IC_NA_1	Interrogation command
101	C_CI_NA_1	Counter interrogation command
102	C_RD_NA_1	Read command
103	C_CS_NA_1	Clock synchronization command
105	C_RP_NA_1	Reset process command
107	C_TS_TA_1	Test command with time tag CP56Time2a
110	P_ME_NA_1	Parameter of measured value, normalized value
111	P_ME_NB_1	Parameter of measured value, scaled value
112	P_ME_NC_1	Parameter of measured value, short floating-point number
113	P_AC_NA_1	Parameter activation
120	F_FR_NA_1	File ready
121	F_SR_NA_1	Section ready
122	F_SC_NA_1	Call directory, select file, call file, call section
123	F_LS_NA_1	Last section, last segment
124	F_AF_NA_1	Ack file, ack section
125	F_SG_NA_1	Segment
126	F_DR_TA_1	Directory
127	F_SC_NB_1	Query Log, Request archive file

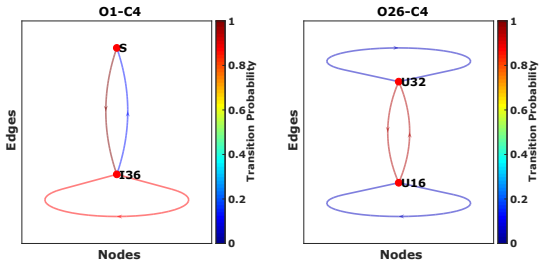

Figure 12: Two of the simplest expected communication patterns in our datasets. The left image shows a primary communication pattern where an outstation transmits monitored data via I-format APDUs and the control server periodically acknowledges via S-format APDU. Similarly, the image on the right shows an ideal secondary (redundant) connection where the outstation and control server exchange U_{16} and U_{32} APDUs (keep alive messages followed by their acknowledgment).

after investigating the cause of these repeated APDUs, we found that this was due to packet re-transmissions at the TCP layer and not unexpected behavior of the endpoints.

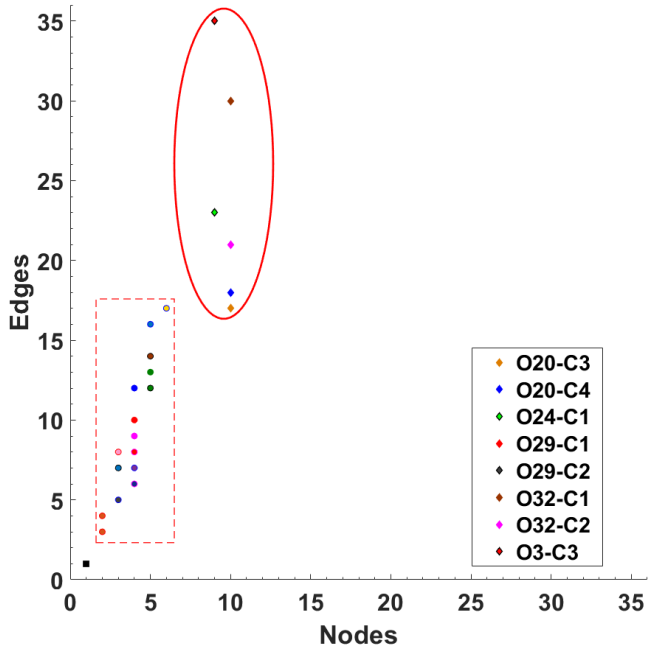

Figure 13: Size of Markov chains for all connections (nodes and edges). All abnormal secondary connections share a single point at (1,1). All connections in the ellipse have "I_{100}" (interrogation command) while those in the rectangle do not

We now study the diversity of Markov chains inferred. Fig. 13 shows the size (in terms of nodes and edges) of the Markov chains inferred in all connections. We can clearly see three clusters: The first cluster has only one node and one edge—the point (1,1); the second cluster is captured by the connections within the square; and the third cluster is captured by the connections within the ellipse.

Fig. 14 shows how all the connections in point (1,1) of Fig. 13 look like: a sequence of repeated U_{16} messages sent by the server without the corresponding acknowledgement U_{32} from the outstation. These are precisely the connections we illustrated in Fig. 9, where U_{16} messages are ignored, and instead the Outstation resets the TCP connection. Connections C2-O28, C2-O24, C1-O7, C1-O9, C1-O6, C1-O8, C1-O35, C2-O30 (cluster 0 in the previous subsection), C1-O15, and C1-O5 all fall into this category.

The other two clusters (square and ellipse) have a variety of Markov chains, but the size of the Markov chain in the connections in the ellipse had much higher number of edges than the others. After inspecting all connections in the ellipse, we observed a command that is not present in any of the other clusters; the interrogation command (an I-format message with typeID 100). I_{100} requests the outstation to send all of the IOAs it monitors, so it basically sends a lot of information, resulting in a variety of new (previously unseen) I types that do not report back to the control center periodically.

Fig. 15 shows one of the communication patterns that included I_{100} (one of the connections from the cluster in the ellipse from Fig. 13). The Markov chain in this figure shows how the server initiates data transfer by sending U_1 `STARTDT act`, which is then

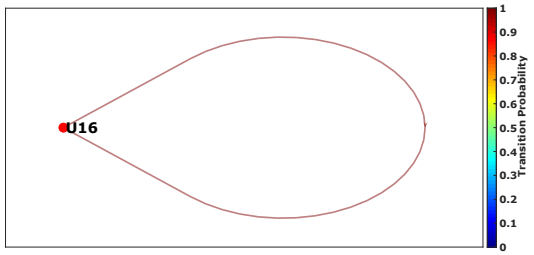

Figure 14: Simplest, but abnormal communication pattern in our datasets. Abnormality due to missing U32 APDUs (acknowledgements to keep-alive messages).

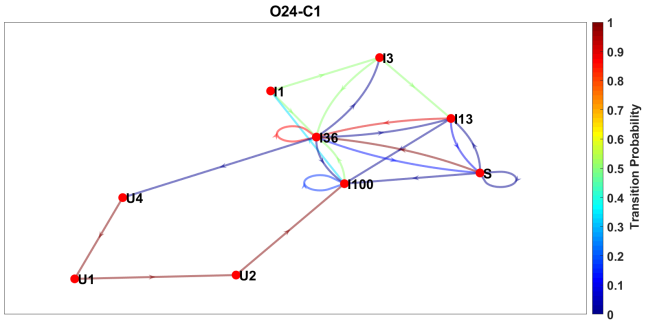

Figure 15: A connection from the cluster in the ellipse. After a request to start sending I-Format APDUs U_1 and the ack U_2 the first I-Format APDU sent is an I_{100} interrogation command from the server, afterwards, the outstation starts transmitting regular I-format APDUs such as I_{13}, I_{36}.

acknowledged with a U_2 STARTDT con from the outstation. The server then sends an interrogation command I_{100}, which instructs the outstation to report all of its field devices and their corresponding measured values.

According to the IEC 104 standard, I_{100} is sent by the control server whenever the server starts the request for I-Format messages, which happens in the following three conditions: (1) on a newly-established connection, (2) on a connection switch from secondary to primary, and (3) operator or program-initiated. Condition (2) explains why most of the connections in the ellipse in Fig. 13 are in pairs (e.g, O20 connecting to both C3 and C4, or O29 connecting to both C1 and C2). An example of these switchovers is illustrated in Fig. 16.

We found the interrogation command also interesting from a cybersecurity perspective. In the most recent cyberattacks on the power grid of Ukraine [7], the attackers developed a malware called Industroyer [2], which targeted IEC 104 networks. Once a TCP connection gets established between a control server and an outstation, Industroyer would start the *ICS reconnaissance phase* by sending IEC 104 packets iteratively to the target ADSU address and IOAs, attempting to discover as IOAs. Instead of iteratively discovering IOAs, a single I_{100} interrogation command would have allowed Industroyer to accomplish the same goal.

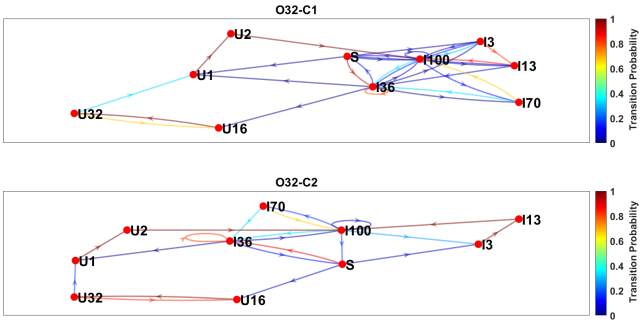

Figure 16: Markov chains showing a switchover between servers C1 and C2, as indicated by having keep-alive messages in secondary connections (U_{16} and U_{32}) and then the initiation of a primary connection, as indicated by U_1 and U_2 followed by I_{100} and then multiple I-format messages.

So far we have figured out that the point (1,1) in Fig. 13 represents secondary connections that reset every attempt by the server to establish a backup connection, and all the connections in the ellipse of Fig. 13 have the interrogation command I_{100}. We now turn our attention to the cluster of connections within the square of Fig. 13. By looking at the Markov chains of these connections, we found a variety of behaviors, summarized in Table 6.

Table 6: Outstation Classification

Type	Description
1	No secondary connection and I-format only
2	With secondary connection and U16&U32
3	U-format only
4	I-format only to both servers
5	Single server with both I and U formats
6	With secondary connection I-format and U16 only

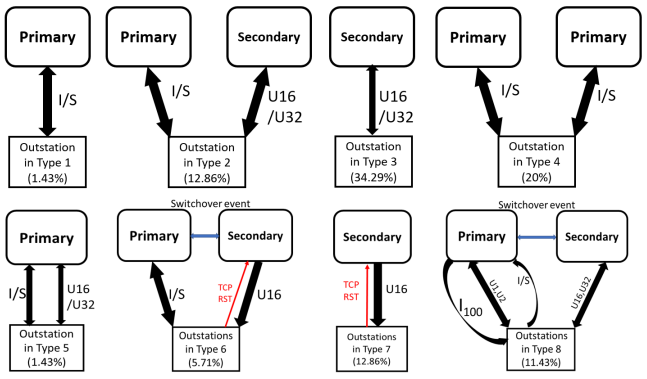

Figure 17: Outstation simplified Interaction for each type in Table 6 and previously described at point (1,1) as Type 7, and in the ellipse as Type 8.

Type 1 outstations are those that only have active connection to its primary control server (the one sending I-Format messages)

but they do not have redundant connection to a secondary control server (the backup connection sending keep-alive messages).

Type 2, on the other hand, is an ideal situation specified in the IEC 104 standard, where each outstation has a TCP connection to its primary server for sending I-format messages while having a secondary connection exchanging periodically keep-alive messages to its backup server.

Type 3 outstations are simply redundant equipment that transmit only U messages. We found this configuration in newer substations (such as S10 in Fig. 6, which has 14 RTUs) where for example, each generator is monitored by two redundant RTUs (outstations), one RTU talks to two control servers (e.g., O10 talking to C3 and C4) while the redudntant RTU simply sends keep-alive messages (O11).

Type 4 consist of only one outstation which has only one active TCP connection at any given time sending I-format messages. This outstation could be classified as a Type 1 connection, but the only difference is that between datasets, it seems the connection switched over to the other server, so it is the only "Type 1" connection that we saw sending I-format messages to both servers.

Type 5 also consists of only one outstation that behaves in a unique fashion: it is the only connection where in the middle of I-format message transmissions, we see a keep-alive/test-connection request (we only see keep-alive request in secondary connections). To examine what happens in this connection, we looked at the IEC 104 standard and saw that when the timer T_3 expires, a keep-alive message will be sent (regardless if it is a primary or secondary connection), and upon further inspection, we found several intervals between I-format messages in this connection that last over 20 seconds (which is larger than the default T_3 timer, thus forcing a keep-alive message). Looking for an explanation for these long intervals without I-messages, we found that the cause of transmission (COT) of all I-format messages was spontaneous (`"Spont"`) which means the I-format messages are sent only after certain thresholds in the measured values are reached (if the measured values do not change enough to meet these thresholds, the values will not be sent over the network), so our hypothesis for these long time-intervals, is that the outstation was configured with large thresholds for spontaneous transmissions. This hypothesis was confirmed with our contact at the utility company who mentioned the outstation sometimes showed stale data in the control room. This is again one of the problems with federated SCADA systems, where unless there is regulation mandating very detailed standards, the individual power companies can select their own configuration parameters, which sometimes lead to sub-optimal performance in the operation of the system.

Finally, Type 6 are outstations O5 and O8. It appeared that O5 and O8 send I-messages to either C1 or C2, whichever is active, and as the other server (inactive) tries to establish a redundant connection, O5 and O8 would refuse the connection, hence only see U_{16} message instead of the expected U_{16} & U_{32} pair.

Fig. 17 summarizes our analysis. In short we can see that most of the outstations (34.3%) we analyze are in type 3, which means that they are backup outstations, serving as reliable connections in case the primary outstation stops responding. Type 3 is very similar to Type 7, but with the difference that type 7 outstations have the misconfiguration about resetting the TCP connection, fortunately Type 7 backup outstations are just a fourth of all the backup outstations. The second most common type is type 4, which means that these outstations changed their connections to the server among our different packet captures (this type is related to Type 8 outstations, where we actually see the server switch-over during our packet capture); therefore we can conclude that a rotation from primary to secondary servers is a common occurrence.

Our results satisfy Hypothesis 4, as we found small state machines useful in identifying a fixed number of communication profiles between control servers and outstations. The next step is to analyze the semantics of the payload.

6.4 Physical Measurements

We are finally ready to explore Hypothesis 5, and determine if through a network tap, we can create profiles from the physical system under control; i.e., to understand the semantic-nature of the information exchanged in the network. Recall that each ASDU is identified by a specific typeID which defines the data format (floating point, normalized, etc.), and that IEC 104 supports 54 ASDU typeIDs, illustrated in Table 5. Out of these 54 typeIDs, only 13 were observed in all our datasets. Table 7 shows distribution for each of these 13 typeIDs. As seen, the most transmitted ASDU typeID were I_{36} (measured value, short floating point with time tag) and I_{13} (measured value, short floating point without time tag). These two types of typeIDs represent almost all of the ASDUs exchanged in the power grid (97%).

Table 7: Observed ASDU TypeID Distribution

ASDU TypeID	Percentage	ASDU TypeID	Percentage
I_{36}	65.1322%	I_{103}	0.0011%
I_{13}	31.6959%	I_{30}	0.0005%
I_9	2.6960%	I_{70}	0.0005%
I_{50}	0.2330%	I_{31}	0.0005%
I_3	0.1427%	I_1	0.0004%
I_5	0.0893%	I_7	0.00004%
I_{100}	0.0080%		

Table 8: ASDU TypeID and Physical Measurement

ASDU TypeID	Transmitting Station Count	Physical Symbols Reported
I_{13}	20	I,P,Q,U,Freq
I_{36}	13	I,P,Q,U,Freq
I_{100}	9	Inter(global)
I_3	6	P,Q,U,Status(0,1,2)
I_{31}	4	Status(0,2)
I_{50}	4	AGC-SP
I_1	3	Status(0)
I_{103}	3	–
I_{70}	2	–
I_5	1	–
I_9	1	–
I_7	1	–
I_{30}	1	–

Legend: I=Current; Q=Reactive Power; P=Active Power; U=Voltage; Freq=Frequency; Inter=Interrogation; AGC-SP=AGC Set point; -=Unspecified.

Table 8 summarizes all observed typeIDs with their corresponding physical measurements (e.g., current, power, voltage, frequency, etc.), and a few typeIDs for which we were not able to identify their semantic value.

In power systems there are some physical quantities that controllers need to keep constant at a given value (e.g., frequency and voltages) while other physical quantities are allowed to change based on consumer demand and the associated response in generation (current and power). We performed a normalized variance analysis of each of these time series in order to identify "interesting" events (events where one variable was changing more than usual).

Our first finding was in the analysis of power, as we can see power fluctuations in Fig. 18, bottom plot. This was a case of "unmet load" caused by a failure, it means that there was a lost electric load at some point, causing the frequency of the power grid to increase because there was more electric generation than electric load. The system operator then needs to ask generators to reduce their production of electricity in order to stabilize the system via AGC messages; once the electric load is reconnected to the power grid, the generation is ramped up again. These sequence of AGC commands and their effect can be seen in Fig. 19.

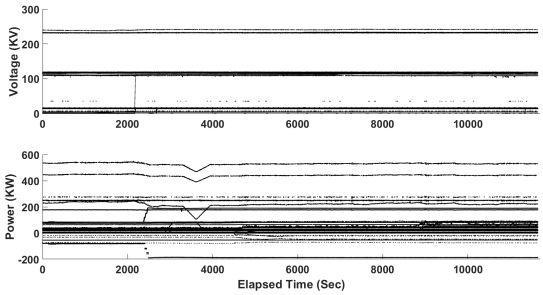

Figure 18: Voltage and active power fluctuations

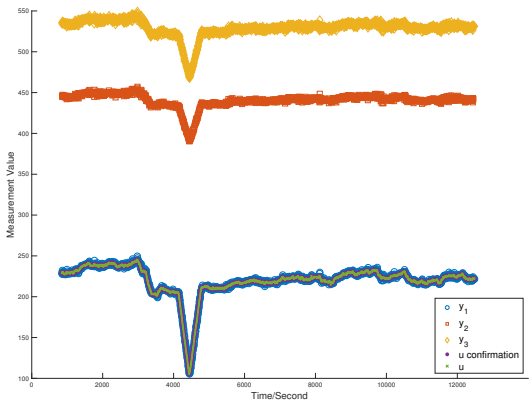

Figure 19: The bottom time-series is the AGC control commands, and the top two series show how generators react to the control actions.

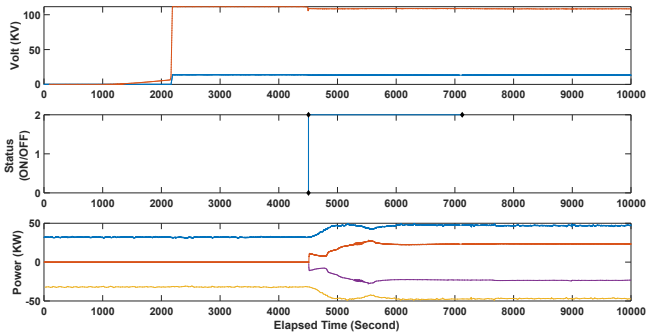

Figure 20: The top plot shows the voltages (at the input and output of a step-up transformer) as the generator was becoming active (ramping up), the middle plot shows a change of status in a circuit breaker that connects the generator to the power grid, and the bottom plot shows the power fluctuations in power once the generator is connected to the grid.

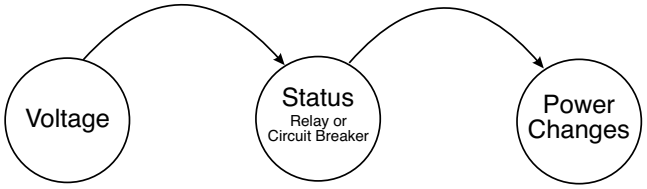

Figure 21: Signature of Power System Behavior

Another interesting finding was when we measured voltage values. Fig. 18 (top) shows that most voltages are within the nominal values; however, we can see that one of the time series jumps from 0kV to about 120kV. This is the signature of when a generator is put online. Connecting a generator to the power grid requires the generator to be synchronized to the grid. The measured voltages have a spike from 0 to its nominal value (a value around 130 kV) indicating that the generator is ready to be connected to the network. Then the operator takes some time to get the generator ready to connect it to the network (i.e. equalizing its frequency and phase to the network's). When it is connected, it starts to deliver active power to the power system, the reactive power can show a positive or negative value since it depends on the voltage needs, so the generator can consume or produce reactive power. We can see these steps in Fig. 20: during the generator ramping up duration, both active and reactive power would remain unchanged until a *Status* (middle time-series) changed from 0 to 2, meaning that it closes the circuit breaker to connect the generator to the power grid. We can create a state machine representing these relationships in Fig. 21, and this machine can be used to identify whether or not future activation of substations follow this expected pattern, or if power variations are justified or not.

In summary, by performing DPI and extracting physical values from these network packets, we can identify anomalies in the physical world, and also create signatures of expected normal behavior (e.g., Fig. 21). We expect that these types of measurements might be useful in future Security Operation Centers (SOCs) for

the power grid, where indicators of cyber-attacks can be correlated with abnormal behavior in the physical world.

7 CONCLUSIONS

While there is a growing interest in the research community in SCADA networks, most of the related work relies on simulations and testbeds [14, 17, 20, 23, 25, 26], and has not analyzed real-world deployments. The closest work to ours is the work of Formby et al. [10] who measured and characterized the TCP connection dynamics of an electric distribution system. In contrast, our paper looks at the **bulk** power system, which is a more fundamental part of the operation of the grid. In addition, our paper also performs deep-packet inspection of the SCADA protocol. As far as we are aware, our paper is the first to take a look at a real-world SCADA network in the bulk power grid, and in addition perform a deep-packet inspection analysis of this network.

In addition, we discussed the challenges for not having all equipment is under the same administrative domain. We also profiled and identified the reasons of the complexity of some connections, as well as the real-world measurements of physical values and their behavior during operation.

As more critical infrastructure systems migrate from dedicated serial communications to Internet-compatible networks, we need to start investigating these new networks, their expected performance, and identify unusual behaviors. In addition, the rising threat of cyberattacks like those who created blackouts in Ukraine [7] should motivate more researchers to help develop defenses for these networks. In particular, the Industroyer malware [2], which helped create a blackout in Ukraine, leveraged IEC 104 to send false control commands to substations. In future work we plan to take a look at how to create white lists that correlate cyber (e.g., Markov networks) and physical (time-series analysis) network measurements to identify suspicious activities in these networks.

ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-19-2-0010, and by NSF CNS 1929406. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

REFERENCES

[1] Ross Anderson and Shailendra Fuloria. 2010. Security economics and critical national infrastructure. In *Economics of Information Security and Privacy*. Springer, 55–66.
[2] ESET Anton Cherepanov. 2017. WIN32/INDUSTROYER A new threat for industrial control systems. https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32_Industroyer.pdf
[3] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras. 2012. A first look into SCADA network traffic. In *Proceedings of the 2012 IEEE Network Operations and Management Symposium*. IEEE, Maui, HI, USA, 518–521. https://doi.org/10.1109/NOMS.2012.6211945
[4] Philippe Biondi and the Scapy community. [n. d.]. https://scapy.net/
[5] Marco Caselli, Emmanuele Zambon, Johanna Amann, Robin Sommer, and Frank Kargl. 2016. Specification mining for intrusion detection in networked control systems. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 791–806.
[6] Webstore International Electrotechnical Commission. 2006. Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles. https://webstore.iec.ch/publication/3746
[7] MICHAEL WALSTROM University of Washington DONGHUI PARK, JULIA SUMMERS. 2017. Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks. https://jsis.washington.edu/news/cyberattack-critical-infrastructure-russia-ukrainian-power-grid-attacks/
[8] Sandro Etalle, Cliord Gregory, Damiano Bolzoni, and Emmanuele Zambon. 2013. Self-configuring deep protocol network whitelisting. *Security Matters* (2013).
[9] David Formby, Sang Shin Jung, John Copeland, and Raheem Beyah. 2014. An Empirical Study of TCP Vulnerabilities in Critical Power System Devices. In *Proceedings of the 2Nd Workshop on Smart Energy Grid Security (SEGS '14)*. ACM, New York, NY, USA, 39–44. https://doi.org/10.1145/2667190.2667196
[10] David Formby, Anwar Walid, and Raheem Beyah. 2017. A Case Study in Power Substation Network Dynamics. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 1, Article 19 (June 2017), 24 pages. https://doi.org/10.1145/3084456
[11] Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. https://doi.org/10.1080/14786440109462720 arXiv:https://doi.org/10.1080/14786440109462720
[12] Cyril Goutte, Peter Toft, Egill Rostrup, Finn ÃÉ. Nielsen, and Lars Kai Hansen. 1999. On Clustering fMRI Time Series. *NeuroImage* 9, 3 (1999), 298 – 310. https://doi.org/10.1006/nimg.1998.0391
[13] Andy Greenberg. 2019. *Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers*. Doubleday.
[14] Ersi Hodo, Stepan Grebeniuk, Henri Ruotsalainen, and Paul Tavolato. 2017. Anomaly Detection for Simulated IEC-60870-5-104 Trafiic. In *Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17)*. ACM, New York, NY, USA, Article 100, 7 pages. https://doi.org/10.1145/3098954.3103166
[15] Celine Irvene, Tohid Shekari, David Formby, and Raheem Beyah. 2019. If I Knew Then What I Know Now: On Reevaluating DNP3 Security using Power Substation Traffic. In *Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop*. 48–59.
[16] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *ACM Comput. Surv.* 31, 3 (Sept. 1999), 264–323. https://doi.org/10.1145/331499.331504
[17] Chih-Yuan Lin and Simin Nadjm-Tehrani. 2018. Understanding IEC-60870-5-104 Traffic Patterns in SCADA Networks. In *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security (CPSS '18)*. ACM, New York, NY, USA, 51–60. https://doi.org/10.1145/3198458.3198460
[18] Kelvin Mai, Xi Qin, Neil Ortiz Silva, and Alvaro A Cardenas. 2019. IEC 60870-5-104 Network Characterization of a Large-Scale Operational Power Grid. In *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 236–241.
[19] Chen Markman, Avishai Wool, and Alvaro A Cardenas. 2018. Temporal Phase Shifts in SCADA Networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. ACM, 84–89.
[20] Peter Maynard, Kieran McLaughlin, and Berthold Haberler. 2014. Towards Understanding Man-in-the-middle Attacks on IEC 60870-5-104 SCADA Networks.. In *ICS-CSR*.
[21] Neil Ortiz. 2020. https://github.com/Cyphysecurity/IEC104_Parser.git
[22] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53 – 65. https://doi.org/10.1016/0377-0427(87)90125-7
[23] Siddharth Sridhar and Manimaran Govindarasu. 2014. Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid* 5, 2 (2014), 580–591.
[24] Robert L. Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (01 Dec 1953), 267–276. https://doi.org/10.1007/BF02289263
[25] Robert Udd, Mikael Asplund, Simin Nadjm-Tehrani, Mehrdad Kazemtabrizi, and Mathias Ekstedt. 2016. Exploiting Bro for Intrusion Detection in a SCADA System. In *Proceedings of the 2Nd ACM International Workshop on Cyber-Physical System Security (CPSS '16)*. ACM, New York, NY, USA, 44–51. https://doi.org/10.1145/2899015.2899028
[26] David I Urbina, Jairo Alonso Giraldo, Nils Ole Tippenhauer, and Alvaro A Cárdenas. 2016. Attacking Fieldbus Communications in ICS: Applications to the SWaT Testbed.. In *SG-CRC*. 75–89.
[27] Alfonso Valdes and Steven Cheung. 2009. Intrusion monitoring in process control systems. In *2009 42nd Hawaii International Conference on System Sciences*. IEEE, 1–7.
[28] C. Wressnegger, A. Kellner, and K. Rieck. 2018. ZOE: Content-Based Anomaly Detection for Industrial Control Systems. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 127–138. https://doi.org/10.1109/DSN.2018.00025
[29] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang. 2013. Intrusion Detection System for IEC 60870-5-104 based SCADA networks. In *2013 IEEE Power Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESMG.2013.6672100