Title

IEC 60870-5-104 Network Characterization of a Large-Scale Operational Power Grid

ISBN

9781728135083

Authors

Mai, Kelvin
Qin, Xi
Silva, Neil Ortiz
et al.

Publication Date

2019-05-19

DOI

10.1109/spw.2019.00051

Peer reviewed

IEC 60870-5-104 Network Characterization of a Large-Scale Operational Power Grid

Kelvin Mai*, Xi Qin†, Neil Ortiz Silva†, Alvaro A. Cardenas†
*University of Texas at Dallas †University of California Santa Cruz

***Abstract*—Modern SCADA systems are interconnected with one or more industrial network protocols such as DNP3, Modbus/TCP, Ethernet/IP, and IEC 60870-5-104 (IEC 104). IEC 104 is a particularly important protocol because it is one of the network protocols used for Automatic Generation Control (AGC), which is the algorithm that maintains electric power balance across large geographical areas. In this work, we focus on an empirical study and observation of a real-world, large scale IEC 104 power network.**

I. Introduction

Many critical infrastructures such as power systems have existed for over a century. It is only in the past two decades that the way to exchange information between different parts of the system has migrated from serial communications to IP compatible networks. Modern Supervisory Control and Data Acquisition (SCADA) systems are interconnected though a variety of IP compatible industrial protocols such as DNP3, Modbus/TCP, EtherNet/IP, ICCP, and IEC 60870-5-104 (IEC 104).

While there is recent interest in obtaining and analyzing the behavior of industrial networks for security purposes [1]–[3], accessing real-world industrial systems is difficult because of the sensitivity of their operations. Entities who manage these critical infrastructure systems are generally conservative in their security posture and do not allow external parties to easily access their networks; therefore most of the data analyzed by previous work represents SCADA systems in small geographical areas, including an energy monitoring system in a college campus [1], or a single distribution substation [4], [5]. In contrast, in this paper we start analyzing the network of the largest SCADA system considered in the literature—as far as we are aware of.

In particular, we study the supervisory control of a power grid using IEC 60870-5-4 (IEC 104) to monitor and balance power systems through the Automatic Generation Control (AGC) algorithm. To our best knowledge, this paper presents the first measurement study of a real-world IEC 104 network used for coordinating AGC in a live system. All previous studies of IEC 104 have been done either with emulated networks or testbeds [6], [7].

Our goal in this paper is to start the analysis of IEC 104 networks in real operational systems and to profile their normal and expected behavior so that in the future we can use them for anomaly or attack detection. While anomaly detection in general information technology (IT) networks tend to have high rates of false positives due to the dynamic nature, most industrial networks are more stable and predictable, with machines talking repeatedly with other machines, and less human activities. Our measurement study offers some important lessons for building accurate network security monitoring systems in these industrial networks: (1) Because these are federated networks (different power companies own and operate different equipment) they reside under different security domains and have different configurations, which complicates building an anomaly detection system as different configurations will lead to different network behavior; (2) in some of our datasets we were unable to see reports from all field devices, and we discovered commands like the interrogation command in IEC 104, which can help us enumerate all the field devices in a given electric substation, (3) while the protocol IEC 104 specifies a wide range of parameters (e.g., ASDU types), in our observations, we only see roughly 10% of these values being used; this big reduction in the number of the type of packets the protocol specifies, and the ones that are actually used, can help us create better white lists to prevent rogue malicious new commands from damaging the network.

The paper is organized as follows, in section II we provide the background on power systems and IEC 104 necessary to understand the measurements in this paper, in section III we describe the network where our datasets have been captured, in section IV we describe our findings, and in the last section we conclude the paper.

II. Background

The electrical grid is generally divided in three parts: generation, transmission, and distribution. Generation and transmission represent the *bulk* of the power grid, where most of the electric power is generated and transmitted over long geographical distances; distribution on the other hand focuses on smaller geographical areas such as cities. In the bulk electric system there are several transmission and generation utilities (called agents) and an operator which coordinates all of them to maintain the reliability and efficiency of the transmission system. These operators are usually called Independent System Operators (ISO) or Regional Transmission Operators (RTOs), and they coordinate the power system of whole countries (e.g., in Europe) or several states (e.g., in the U.S.).

ISOs and RTOs coordinate the operation of large-scale power systems across multiple electric utility companies and generators. No other organization has the ability to influence the power system operation at a larger scale. The central part for keeping power balance in the bulk system is an algorithm called Automatic Generation Control (AGC), which is responsible for matching electric generation to the load instantaneously across multiple geographical regions. AGC asks different agents to ramp up or down their electric generation

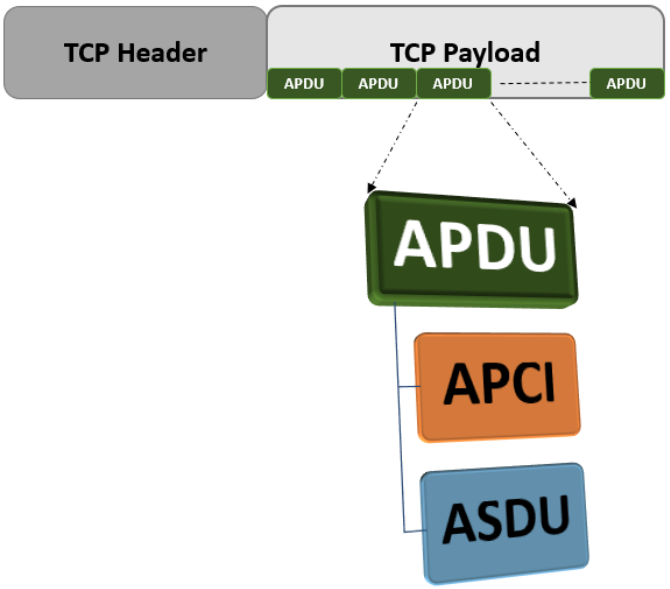

Fig. 1. IEC 104 APDU as part of TCP payload

to maintain the adequate reliability of the overall bulk power system. The network protocol IEC 104 is a popular standard used to relay information back and forth between different substations and the ISO or RTO. IEC 104 is generally used to collect relevant electrical measurements such as frequency, current, voltages, etc., from multiple agents (electric transmission companies) and also to send AGC control commands to generators in different areas.

IEC 104 is an IP-compliant network protocol that is built on top of the previous serial communications standard IEC 60870-5-101 (IEC 101) [8]. IEC 101 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 a control station (e.g., SCADA centers) and outstations via a permanently connected communication link over the telephone network i.e., modem circuit. Then in 2000, as an extension to IEC 101, IEC 104 was defined by the IEC to transport IEC 101 telecontrol messages over TCP using port 2404. That is IEC 104 encapsulates IEC 101 telecontrol messages into an Application Protocol Data Unit (APDU) which is transmitted as part of a TCP payload, as illustrated in Fig. 1. Each TCP payload can carry multiple APDUs up to the TCP maximum segment size (MSS).

APDU fields consist of two parts (as illustrated in Fig. 2): (1) the Application Protocol Control Information (APCI) which functions as a header, and (2) the Application Service Data Unit (ASDU) which contains IEC 101 application data i.e., telecontrol messages.

APCI is a 6-octet header that includes a start octet (68H), followed by 1-octet length, and 4-octet control fields. The least significant two bits of the first control field octet determines the type of telecontrol message, which can be either I-Format (00b), S-Format (01b), or U-Format (11b). An I-Format APCI carries sensor and control information, and it is

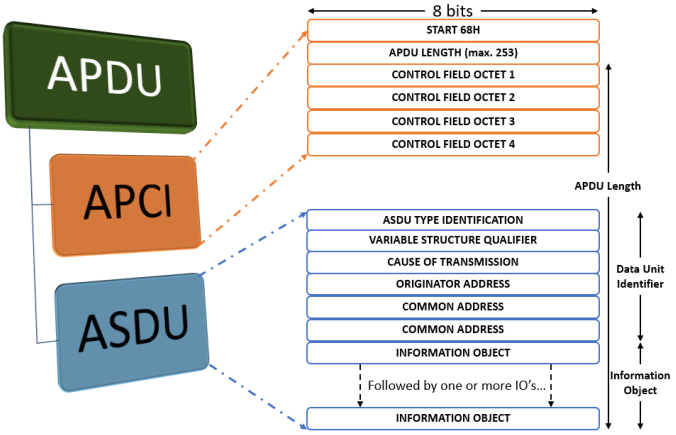

Fig. 2. IEC 104 APDU octets structure

always followed by an ASDU. An S-Format APCI is used to send sequence number acknowledgements, and the U-Format APCI is used to send connection keep-alive messages.

ASDU is a variable length data structure that consists of a fixed-length Data Unit Identifier (DUI) and a variable length Information Object (IO) as illustrated by Fig. 2.

Finally, IEC 104 also defines four timers, their functional descriptions are as follows:

- T_0: time-out of this time will trigger a SYN request.
- T_1: time-out of this timer will trigger an active close request on an established connection. In addition, T1 time-out event on a controlling station may automatically trigger a new redundant connection and an automatic connection switch-over.
- T_2: expiration of this timer will cause the receiver to send a S-Format with a received sequence number (ACK message) which is an application-level acknowledgement for having received a specific number of APDUs.
- T_3: expiration of this timer indicates that there has been no IEC 104 messages received from the other side for a given open TCP connection, i.e. inactivity. A reception of any message type, I-Format, S-Format, or U-Format will reset this timer.

III. IEC 104 Datasets & Network Setting

Our datasets were captured from an operator controlling the bulk power system that serves a population of about 40 million and cover a large geographical area. This power generation system includes two control centers (main and backup) with 4 control servers C1, C2, C3, and C4 (two servers in each physical control center), 22 substations (S1-S22) (including 18 generation substations) with 38 remote terminal units (RTUs) (or as the IEC 104 protocol calls them "outstations" O1-O38). In total, these RTUs control nearly 900 field devices or intelligent electronic devices (IEDs) in the power grid. The 4 control servers work pairwise in a load balancing scheme and each pair controls a number of designated substations. Each substation has one or more outstations. For example, substation 2 has two outstations "O2" and "O36". Each outstation then

collects values from various field devices with measurements including voltage, current, etc.

Our datasets were captured on 5 different dates and times. Control servers and outstations are identified via IP address, whereas substations are identified via a specific geographic location name. Each field device IED is identified by a combination of Information Object Address (IOA) and ASDU address (according to the IEC 104 standard). Table I summarizes our packet captures.

TABLE I
DATASET DETAILS

Dataset	Number of Packets	Packet/sec	Duration (H:M:S)
1	91,427	231.1	0:06:36
2	1,537,049	236.5	1:48:18
3	1,599,322	233.3	1:54:15
4	1,869,741	234.4	2:12:55
5	2,757,728	236.8	3:14:07

IV. ANALYSIS

A. TCP flow Dynamics

We start our analysis with a basic view of the network flows, where a TCP flow is defined by the 4-tuple <srcIP, srcPort, dstIP, dstPort>. Our first goal is to understand the network behavior with the magnitude and direction of network traffic flows. Fig. 3 shows a network flow chord diagram that illustrates the network topology (where devices talk to each other) and complements this with details of the number of packets seen in each flow [9].

This type of network flow visualization provides a quick overview of how active or inactive, and in which direction a particular flow is to be expected. For example, the flow from outstation O4 to control server C3, indicated by the fat purple ribbon in Fig. 3 shows that O4 is the most active outstation. In contrast, outstation O28 is represented by two skinny ribbons (to C1 and C2) showing that it is the outstation that sends the least amount of packets in the network. To understand why there is such a disparity between these outstations, we analyzed the type of data sent by them and the associated field devices reported by these two outstations. For example, O4 reported 14 field devices (IEDs) that help control and monitor 4 generators. Moreover, these generators were observed to have reported a number of different physical measurements via I-format APCIs, which consist of the following measurements: active and reactive power, voltages, currents, frequencies, and AGC setpoints. On the other hand, O28 which is also connected to a generator, didn't report any data during our captures (i.e., we saw no IED from this outstation). In fact, over the duration of all five datasets, O4 transmitted over 284K packets to C3. On the other hand, outstation O28 transmitted less than 5K packets to both C1 and C2, and these were all U-Format APCIs which are only used to keep alive the TCP flow, and not for transmitting data (I-format). Hence, from an operational or security aspect, the operator of this outstation O28, needs to understand why O28 did not report any measurements, intentional or not. We reported in fact this anomaly to the power system operator and they responded that in fact this is a known problem they have from that substation; the substation does not report data frequently enough and the values that appear in their SCADA system appear "frozen" for extended periods of time. And while the operator has asked the electric utility that owns the generator to change the threshold values of the spontaneous transmissions of data (i.e., when the outstation notices the value of a variable has changed beyond a threshold of the last reported value and then sends this new info to the SCADA server), the substation owners have not followed these requests. This case study shows one of the advantages of identifying anomalies in what is considered to be "benign" network traffic and fixing them/clarify them before attempting to pursue network security monitoring.

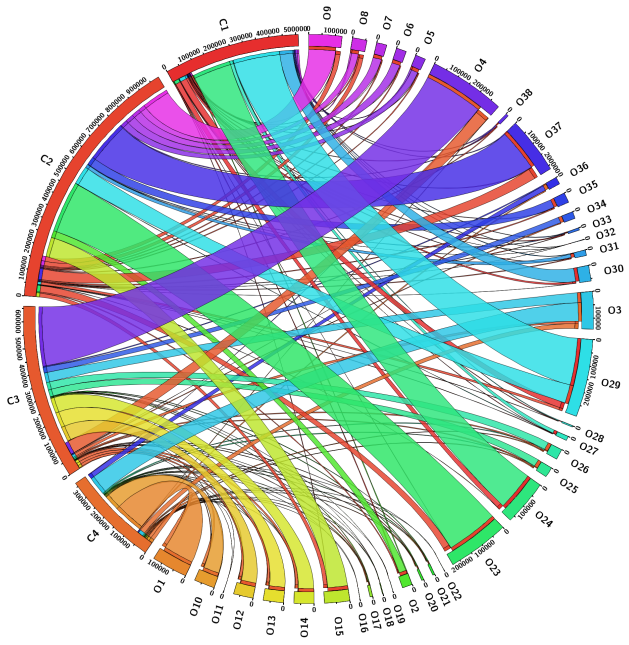

Fig. 3. IEC 104 network flow based on Chord diagram. Each arc segment on the circle perimeter represents an endpoint and the ribbon that connects any two arc segments represents a flow. A small gap (a small white space) between the end of a ribbon and an arc indicates incoming direction, no gap indicates outgoing direction, and each ribbon's width proportionally represents the flow magnitude based on the number of packets seen per flow. Every tick mark, located on the outer edge of each arch segment represents 50K packets.

We now study the duration of TCP flows. We characterize flows in two distinct groups: short-lived flows, and long-lived flows. A short-lived flow must have a matching SYN-FIN or SYN-RST pair in our dataset (that means that the connection started and ended in our data capture). On the other hand, a long-lived flow does not have a matching SYN-FIN or SYN-RST pair (i.e., the flow was active for the whole duration of our data capture, or we saw the beginning or the end of it, but not both). In our

Most long-lived flows on the other hand tend to last the whole duration of our datasets as seen by the 4 horizontal lines that divide long-lived plot (orange color) in Fig. 4. This plot also shows that there were a few outliers (in the 2 ellipses) which had duration considerable less than the dataset duration

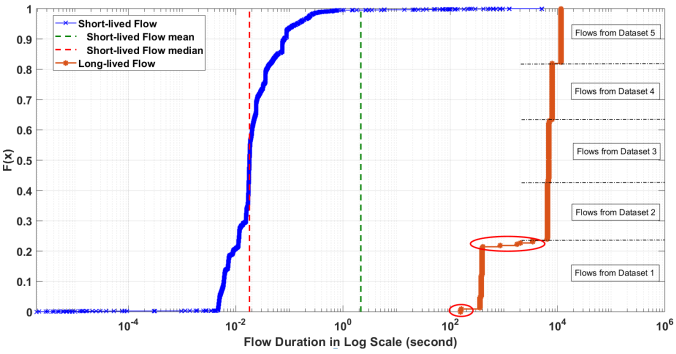

Fig. 4. CDF of TCP flow duration. We can see that 99% of the short-lived TCP flows last less than a second. This is due to RST commands from half of the outstations, which reject any attempt to create a redundant connection from another SCADA server.

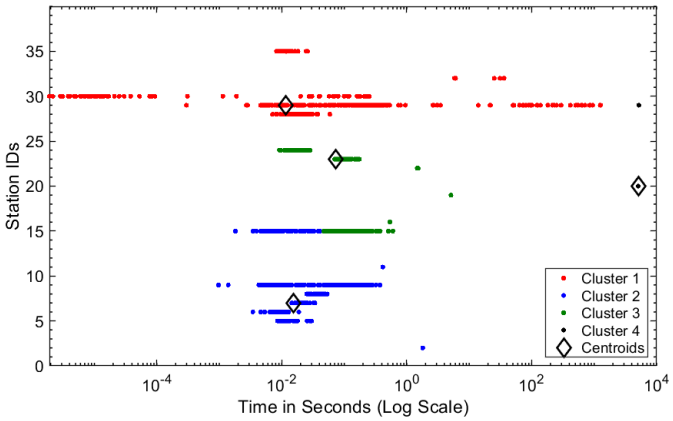

Fig. 6. TCP SYN-FIN Flow Clustered By Duration and Station IDs

because the connection was either started, or terminated (but not both) during our observation period.

Detailed analysis of one of the outliers (from the smaller ellipse) shows that outstation O20 was having two concurrent flows, one each to C3 (active) and C4 (standby). In this setting, C4 and O20, according to IEC 104 standard, must regularly exchange U-format *TESTFR act and con* messages to prevent timer T_1 from being expired, otherwise the flow is terminated. For yet unknown reasons, C4 after exchanging a number of U-format *TESTFR act and con* messages with O20, at one point during the capture, decided to stop sending anymore *TESTFR act*. As a result, O20 sent a FIN-ACK to C4, effectively terminating the flow between O20 and C4.

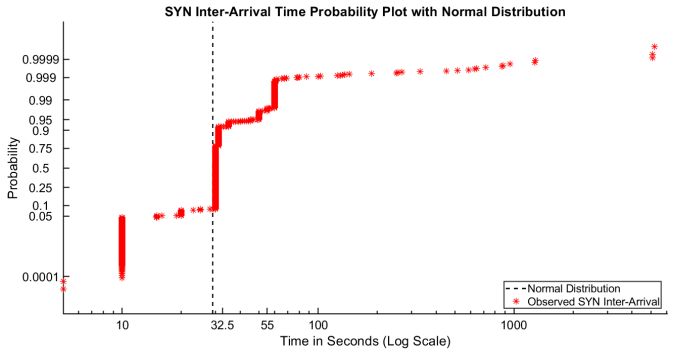

Fig. 5. TCP Flow SYN Inter-Arrival Time

This last point relates to our third question: why did control stations keep sending SYN requests? Recall that IEC 104 defines 4 timers, in which T0's expiration will trigger sending a SYN request. Our empirical study of SYN inter-arrival time suggested that control stations initiated SYN request, on average, about every 30 seconds as shown by Figure 5, and was later confirmed by the balancing authority who indicated that their control stations timer T0 is indeed set at 30 seconds.

As for the final question, which are the outstations that terminated the connections quickly?, Figure 6, a K-mean cluster with K=4 with flow duration as a primary feature while station ID is the secondary feature, shows that from cluster 1, 2, and 3, outstations O30, O29, O24, O23, O15, and O4 to O9 all had sub-second flows.

To complement our analysis of short SYN connections, Figure **??**, a QQ plot, indicated that there are 3 outliers with duration over 5000 seconds, these outliers are outstations O7 and O8 which had TCP flows to C1, and O28 had TCP flow to C2 as shown in Table **??**. We found out that these outliers were flows that had connections switched from one control station to the other during the captured window. Besides these 3 outliers which had long flow duration, nearly 25K other flows had sub-second flow duration. Clearly, this is an abnormal behavior and required additional analysis.

B. APCI Measurement

TABLE II
OBSERVED APCI FORMAT TYPE DISTRIBUTION

Type	Percentage
I-Format	83.5%
S-Format	12%
U-Format	4.5%

After the basic TCP dynamic analysis, our next step was to take a more detailed look of the values exchanged between machines; i.e., do a deep-packet inspection of the fields used by the devices in the network. The header of IEC 104 packets (APCI) can be either I, S, or U-Format, and as Table II shows, I-Format APCIs contribute about 83.5% of our data, while S-Format and U-Format messages contribute 12% and 4.5% respectively. This makes sense as the main goal of the IEC 104 protocol is to exchange data among devices, which is done with I-Format packets; while U and S-Format packets are mostly used for acknowledgement and network signaling. In this subsection we briefly summarize our observations of the signaling S-Format and U-Format packets, and in the next subsection we discuss the I-Format packets that send sensor and control command information.

To protect against loss and duplication of I-Format messages, IEC 104 defines the S-Format APCI which is used by the receiving station to transmit an ACK with the Receive Sequence number (N(R)) back to the sending station. In other words, IEC 104 S-Format and N(R) work similarly to a TCP

ACK and sequence number respectively. In addition, IEC 104 also defines two user configurable variables that control when the receiver should send a S-format with N(R): 'w' and timer 'T2'. 'w' specifies the maximum number of received I-Format APDUs that the receiver should ACK at the latest, and expiration of T_2 immediately triggers sending a S-Format by the receiver regardless of number of APDUs received.

We now take a look at how IEC 104 S-format and N(R) behave in a live system. Our S-Format analysis shows that for most flows with I-format APCI, control servers sent an ACK, on average, after receiving 8 APDUs as shown in Fig. 7. However, observed data also showed that some servers sent S-format for every 1, 2, or 3 APDUs received. Moreover, timer T_2 expiration due to network conditions such as packet loss, appeared to have contributed to the small percentage of N(R) that were less than 8 APDUs.

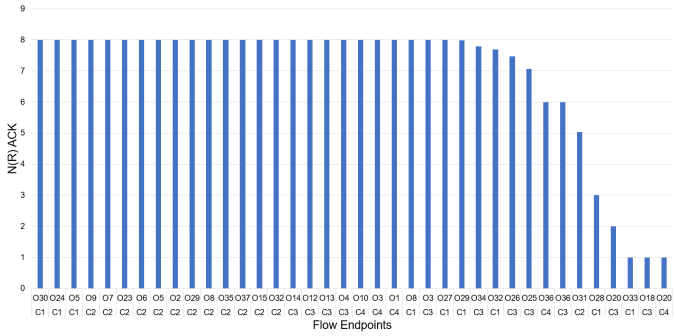

Fig. 7. Most servers send an ACK after 8 I-Format packets; however, some of them are configured to send ACKs more frequently. It is unclear why the server operator chose different parameters for their connections.

We now study U-Format packets, which provide two main functions, (1) connection test (keep-alive) and (2) transmission control. Connection test involves sending a test APDU *TESTFR act* which are confirmed by the receiving station with a *TESTFR con*. Either station may initiate connection test by sending a *TESTFR act* right after T_3 expires. T_3 expires when there has been no messages received from the other station since either the last T_3 reset or last message received, whichever occurred later. We observe that *TESTFR act* APDUs are about 54.4% of all U-Format APDUs. Similarly, *TESTFR con* APDUs are 45.4%.

C. Analysis of I-Format Packets

We now focus our attention to the type of information that is exchanged between machines in this network. Our first observation is that while IEC 104 defines 127 different types of information that can be exchanged (e.g., single-point information, integrated totals, set point commend, bistring of 32 bits, etc.) only thirteen ASDU type identification codes are seen in our datasets, and only three identification codes actually have significant occurrences while the rest of the other ten identification codes have very negligible occurrences in comparison. Table III shows that Type ID "M_ME_TF_1" No. 36 "Measured value, short floating point number with time tag CP56Time2a" is the most frequent code with 65%, followed by "M_ME_NC_1" No. 13 "Measured value, short floating point number" with 32%, then "M_ME_NA_1" No. 9 "Measured value, normalized value" with 3%. The rest of the ASDU type identification codes have less than 1% occurrences. From a security perspective this reduction on the types of values expected can help us create a white list where we would flag as anomalous any code exchanged that is not part of the 13 observed during our training period of time.

TABLE III
OBSERVED ASDU TYPEID DISTRIBUTION

ASDU TypeID	Percentage	ASDU TypeID	Percentage
M_ME_TF_1	65.1322%	C_CS_NA_1	0.0011%
M_ME_NC_1	31.6959%	M_SP_TB_1	0.0005%
M_ME_NA_1	2.6960%	M_EI_NA_1	0.0005%
C_SE_NC_1	0.2330%	M_DP_TB_1	0.0005%
M_DP_NA_1	0.1427%	M_SP_NA_1	0.0004%
M_ST_NA_1	0.0893%	M_BO_NA_1	0.00004%
C_IC_NA_1	0.0080%		

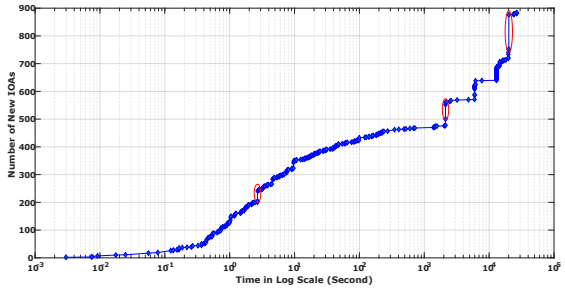

Fig. 8. New IOA discovered over time. Bursty jumps enclosed in ellipses are explained in the text. The above graph suggests that a large number of new IOAs were discovered within a few seconds which is consistent with findings from one of our previous work [10].

The next element within the ASDU structure as shown in Fig. 2 is the Information Object (IO). Each Information Object (IO) has an assigned IO address (IOA), together with the Common Address (CA), they identify a unique field device such as sensor, relay etc. [11]. Creating a white list of the IOAs allowed in the network can be one of the first steps to protect a SCADA network, and during an asset discovery period, the question is how fast can we identify all IOAs active in the network (so that we can alert whenever new IOAs are introduced). Unlike traditional IT network in which new devices are discovered gradually over time, SCADA new devices are expected to be discovered much quicker in part due to periodic machine to machine communications [10]. Fig. 8 shows the number of IOAs discovered over time. We can see that the number of new IOAs observed in traffic grows fairly fast (thus the log-scale) and in addition, there are three jumps, illustrated by the 3 ellipses in Fig. 8.

At the first bursty jump, at about 2.7 seconds, O30 transmitted 37 IOAs to C1 via ASDU typeID M_DP_NA_1:"Double-point information without time tag" with Periodic/Cyclic (Per/Cyl) cause of transmission (COT), and these 37 IOAs had not been transmitted previously since the start of the capture. Prior to transmitting these 37 IOAs, O30 had transmitted 4 other ASDUs of typeID M_ME_TF_1:"Measured value,

short floating point number with time tag CP56Time2a" that included 5 IOAs. Since these 37 IOAs were transmitted via Per/Cyl COT which suggested that these 37 IOAs would be transmitted at a fixed time cycle, i.e. controlled by some sort of timer. As such, we further analyzed this particular type of Per/Cyl COT from O30, and found that O30 transmitted these same 37 IOAs every 10 seconds. Periodic transmission of data as seen by this event can be essential in modeling intrusion detecting system, which have been proposed previously by several papers [1], [2], [6].

In the second jump, just after 30 mins of observation, we found 3 ASDUs reporting a total of 127 IOAs, from which 52 had not been reported previously. To understand this sudden increase in the number of IOAs, we examined the type of packets sent related to this jump. We found outstation O29 transmitting to C1 three I-Format ASDUs of typeID M_DP_NA_1:"Double-point information without time tag" with COT of interrogation procedure (Inrogen). According to the IEC 104 standard, an interrogation command basically asks the outstation to send all the values of all IOs it has; and therefore this accounts for the big jump of previously unseen values. As we found out that not all values are frequently reported and are only seen in the network when the control center sends the interrogation command. This is perhaps a command that should be executed when an intrusion detection system or network monitor is trying to learn the assets available in the network. Details of the actual packet that transmitted 60 IOAs between O29 and control station C1 can be seen in Fig 9.

Unlike previous two bursty events, the third one was caused by rather a rare event from outstation O3. In fact, this type of event from O3 is only observed once in all 5 datasets. At about 6 hours of observation, O3 transmitted a single TCP packet that contained 8 ASDUs. The first ASDU typeID was C_IC_NA_1:"interrogation command" with activation confirmation (actcon) COT, followed by 6 ASDUs of typeID M_DP_NA_1:"Double-point information without time tag" with Inrogen COT, and the last ASDU of typeID C_IC_NA_1 with activation termination (ActTerm) COT. From these 8 ASDUs, 210 IOAs were all transmitted from M_DP_NA_1 ASDUs, out of which 124 had not been reported previously. One may ask how or what might had triggered this rare event to occur? The answer seemed to point to a network condition. About 33 seconds prior to this rare event, C3 sent multiple unsuccessful TCP re-transmission packets to O3, which resulted in O3 terminating the TCP connection with a RST packet. 10 seconds later, C3 attempted to establish another TCP connection to O3 via SYN packet which O3 accepted via SYN-ACK. Soon afterward, data transfer was initiated via a pair of matching *STARTDT act* and *STARTDT con*, and within 279 msec, this rare event occurred.

Most of the I-Format packets are simply outstations reporting their measurements to the control centers, but we found two types of I-Format packets sent from SCADA servers to outstations, namely the C_IC_NA_1 Interrogation command discussed in the previous section, which accounted for 5.6% of the I-Format control messages sent to outstations.

The other 94.4% of the I-Format control messages sent

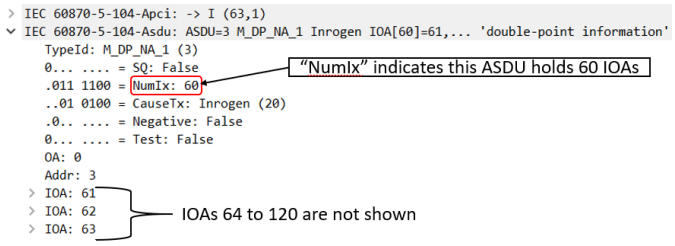

Fig. 9. M_DP_NA_1 ASDU with Inrogen COT (contained 60 IOAs), which contributed to the second bursty jump of Fig. 8

from the control centers to the outstations were AGC control commands; namely ASDU C_SE_NC_1, which is a set point command. This command basically asks a power generator to ramp up or down its electric power generation to match the set point desired by the control center.

D. IEC 104 Flows in Control Direction Sending I Type APDUs

To differ from the concept of TCP flows defined in a four-element tuple in section **??**, we name "IEC 104 flow" as a single-directional connection between one control station and one outstation where IEC 104 APDUs are transmitted. As defined in terminology **??**, *control direction* is from control stations to outstations, and *monitor direction* is from outstations to control stations. We consider I type APDUs sent by control stations as unusual cases, from the following statistics:

E. Preliminary Clustering of IEC 104 Sessions

To differentiate from the concept of TCP flows, we name an "IEC 104 session" as a single-directional connection between one control station and one outstation where IEC 104 APDUs are transmitted. Any session involving transmission of IEC 104 APDUs is considered as a IEC 104 session. We extracted traffic statistics of IEC 104 APDU transmissions for each IEC 104 session and applied clustering algorithms to group sessions with similar traffic patterns into clusters. Our objective here is to use clustering group sessions with the same or similar behavior patterns and to identify all the groups with normal activities. With the identified groups, it's possible to locate the abnormal session(s) in current dataset. Moreover, it's a baseline to monitor and to examine any future IEC 104 sessions. The complete clustering process is shown in Fig.10.

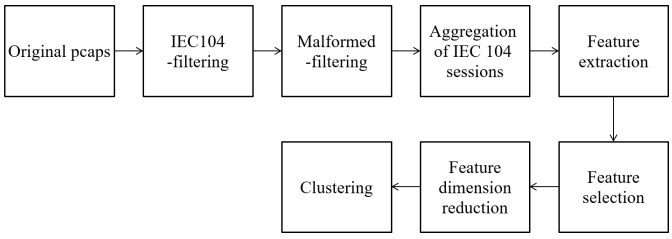

Fig. 10. A complete work flow of IEC 104 session clustering

1) Feature Vector Extraction and Clustering Process:

- dir_i, direction of APDUs being transmitted. 0 represents for the session direction from an outstation to a control station, monitor direction. 1 represents for the session direction from a control station to an outstation, control direction.
- Δt_i, average time interval between two consecutive packets in time series of session s_i. $\Delta t_i = (t_{last}$, epoch time of the latest packet in session s_i - t_{first}, epoch time of the earliest packet in session $s_i) \div N$ total number of packets in session s_i containing IEC 104 APDUs. Time series span all five captures.
- len_i, total payload size in session s_i. Sum of octet lengths in all APDUs transmitted in this session across all five captures. l_j is the octet length of the j_{th} APDU in session s_i which has total J APDUs.
- num_i, number of times when a packet from session s_i is captured or count of packets in session s_i, computed from all five captures.

TABLE IV
MATHEMATICAL REPRESENTATION OF FEATURES IN FEATURE VECTOR

Feature symbol	Mathematical description
dir_i	$\begin{cases} 1, \text{control direction} \\ 0, \text{monitor direction} \end{cases}$
Δt_i	$\frac{t_{last} - t_{first}}{N}$
len_i	$\sum_j^J l_j$
num_i	N

Other features have been considered as well, such as the capture time of the first/last packet per session, sum of IOAs' count and sum of APDUs' count per type (U/S/I). Feature space dimensionality has been reduced from $d = 10$ to $d = 4$. This process has been done via computing the mean and variance of the session distribution under each feature. Features with a relatively large discrimination are chosen to be in the final feature vector representation.

After data extraction based on the above feature vector, we perform our clustering using K-means clustering algorithm [12]. Proper number of clusters K is decided as six via Elbow method, statistically verified through Silhouette score and verified when having the results.

To further reduce feature dimensionality and noise, and also to better visually present the clustering result, we conduct principle component analysis (PCA) [13] on clustering results, which will be presented in Fig.11. The variance retained in two principle components from the dataset is 86%.

2) Clustering Result and Analysis: Unsupervised learning on network traces from such a tremendous real-world SCADA network is a challenging task. The cardinality of the dataset is extremely large as discussed in section IV-E1 and this problem has been remedied through session aggregation. The percentage of packets which may have abnormal behaviors generally take a small portion, which leads to sparse anomaly points to be separated from a large group of normal packets. Furthermore, clustering results must have meaningful clusters, which coordinate the real-world network and reflect issues that are application-specific.

We pick the appropriate cluster number based on the selected feature vector candidate. Usually there's no common or critical standards to pick K, a proper K leads to relatively reasonable clusters which are domain dependent [14]. With the facility of domain knowledge and statistical metrics like Elbow method on both sum of squared error [15] and explained variance [16], and Silhouette scores [17], we pick the most effective cluster number: 5. Clustering results are shown in Fig.11.

In Fig.11, there's only one session which has been classified into cluster 3, an evident outlier. This session was considered as abnormal by the algorithm since it has the largest Δt_i and smallest len_i. Its session direction was from one control station to an outstation. APDUs being sent in this session are U type in testing mode (TESTFR) only. As briefly discussed in section IV-B, one station sends TESTFR-act to the other station under a timer T3 to confirm the connection is on. If no TESTFR-con is received back when T3 expires, then another timer T1 is activated. In this session, one specific control station (C2) sent testing at a much longer time-out (around 430 seconds) to a outstation (O30). Unlike other sessions where control stations send TESTFR-act to outstations and get response with TESTFR-con, O30 never replied C2 with a testing confirmation signal. This behavior has been explained in section **??**, and has been confirmed by our data provider. If one outstation has already established connection of sharing data with one control station, then a second connection request from another control station (usually a SYN packet), will be turned down by a FIN or RST packet from the outstation. In this session, O30 has already established a stable connection with C1 and has sent I type APDUs to C1. When C2 requests a connection with a SYN packet, O30 would send RST package back to C1 every 30 seconds as termination. 30 seconds is the timer T0, which is the timer for control stations sending SYN packets.

The abnormal part in this specific session is that C1 sends TESTFR-act to O30 at a much slower rate ($\sim$ 430 seconds). However in other sessions under the same scenario, the rate is $\sim$ 30 seconds. Because each time control stations will only send a TESTFR-act after a SYNC packet and timer T0 is 30 seconds for sending SYNC packets. Therefore, this session is identified as an outlier because of the unusual testing rate.

Cluster 0 contains all sessions transmitting APDUs from control stations to outstations. All the rest clusters contain all the sessions in the opposite direction.

Cluster 4 contains all sessions with the significantly more packets and larger sized APDUs, which are at the scale of 10e5 and 10e7 of octet lengths, respectively. The most active outstations are: O29 and O1. APDUs transmitted in these sessions have larger size because at least one ASDU type from the following three is involved: NO.31, M_DP_TB_1, Double-point information; NO.36, M_ME_TF_1, measured value, short floating point number; NO.103, C_CS_NA_1, Clock synchronization command. These three types all have a common time tag CP56Time2a, which takes seven octets more. With Cluster 4, the most active sessions at a higher scale of traffic amount can be identified.

Cluster 2 contains the second most active group of sessions.

Cluster 1 contains all the rest sessions from outstations to control stations with less active traffic, compared to sessions in Cluster 2 and 4.

From the above analysis, we have clearly separated clusters and an evident outlier. We need to further extend the current clustering algorithm to a more sophisticated anomaly detection algorithm. Other features such as ASDU types and components of each information object must be included to hierarchically classify each individual cluster.

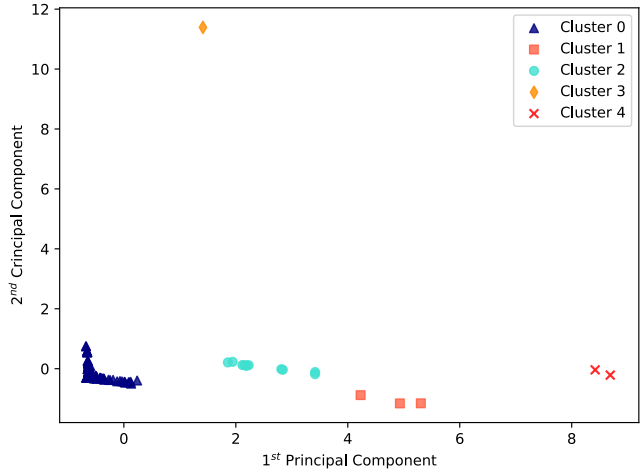

Fig. 11. PCA of clustered IEC 104 sessions with "direction of sessions" feature

V. Conclusion

In this paper, we present the first analysis of a large-scale real-world IEC 104 power grid network.

We have identified unique challenges that arise in industrial networks under different administrative domains, such as the fact that other companies select to reject backup connection requests from the regional authority, and the fact that one of the outstations keeps sending malformed packets but the regional authority cannot reconfigure these devices. As far as we are aware, all previous work measuring real-world industrial networks has focused only on networks by one company under a single administrative domain.

We have also done a detailed summary of the types of behaviors in IEC 104 networks, like identifying AGC control commands, identifying interrogation commands, who send everything an RTU has and thus increases our visibility into the monitored values, and also identified practical timers and mechanisms to keep a connection alive, even when not transmitting I-Frames. Knowing all these parameters and available commands (e.g., interrogation commands) can help us define white lists for intrusion prevention.

For the next phase of our research, we will continue with in-depth IEC 104 packet analysis from which we hope to be able to construct a probabilistic model or other applicable learning models that could be applied to set the baseline of normal activity in the network, and to generate anomaly detection rules with the goal of designing an intrusion detection system for this network.

Acknowledgements

This work was supported by NSF grants CNS-1718848, CMMI-1541199, NIST award 70NANB16H019 and by DHS/AFRL FA8750-19-2-0010.

References

[1] N. Goldenberg and A. Wool, "Accurate modeling of modbus/tcp for intrusion detection in scada systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63 – 75, Jan. 2013.

[2] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, ser. CPSS '15. New York, NY, USA: ACM, 2015, pp. 13–24. [Online]. Available: http://doi.acm.org/10.1145/2732198.2732200

[3] R. R. R. Barbosa, R. Sadre, and A. Pras, "A first look into scada network traffic," in *Proceedings of the 2012 IEEE Network Operations and Management Symposium*. Maui, HI, USA: IEEE, 2012, pp. 518–521. [Online]. Available: https://ieeexplore.ieee.org/document/6211945/

[4] D. Formby, S. S. Jung, J. Copeland, and R. Beyah, "An empirical study of tcp vulnerabilities in critical power system devices," in *Proceedings of the 2Nd Workshop on Smart Energy Grid Security*, ser. SEGS '14. New York, NY, USA: ACM, 2014, pp. 39–44. [Online]. Available: http://doi.acm.org/10.1145/2667190.2667196

[5] D. Formby, A. Walid, and R. Beyah, "A case study in power substation network dynamics," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, pp. 19:1–19:24, Jun. 2017. [Online]. Available: http://doi.acm.org/10.1145/3084456

[6] C.-Y. Lin and S. Nadjm-Tehrani, "Understanding iec-60870-5-104 traffic patterns in scada networks," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, ser. CPSS '18. New York, NY, USA: ACM, 2018, pp. 51–60. [Online]. Available: http://doi.acm.org/10.1145/3198458.3198460

[7] E. Hodo, S. Grebeniuk, H. Ruotsalainen, and P. Tavolato, "Anomaly detection for simulated iec-60870-5-104 trafiic," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17. New York, NY, USA: ACM, 2017, pp. 100:1–100:7. [Online]. Available: http://doi.acm.org/10.1145/3098954.3103166

[8] W. I. E. Commission. (2006) Telecontrol equipment and systems - part 5-104: Transmission protocols - network access for iec 60870-5-101 using standard transport profiles. [Online]. Available: https://webstore.iec.ch/publication/3746

[9] M. Iturbe, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, "Visualizing network flows and related anomalies in industrial networks using chord diagrams and whitelisting," in *VISIGRAPP*, 2016.

[10] A. Srivastav, C. Ortega, P. Ahuja, M. Christian, and A. Cardenas, "Exploratory analysis of modbus and general it network flows in a water scada system," in *Industrial Control System Security Workshop*, vol. 78, 2015.

[11] W. I. E. Commission. (2003) Telecontrol equipment and systems - part 5-101: Transmission protocols - companion standard for basic telecontrol tasks. [Online]. Available: https://webstore.iec.ch/publication/3743

[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[13] K. P. F.R.S., "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*, vol. 2, no. 11, pp. 559–572, 1901. [Online]. Available: https://doi.org/10.1080/14786440109462720

[14] A. K. Jain, "Data clustering: 50 years beyond k-means," vol. 31, pp. 651–666, 06 2010.

[15] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, Dec 1953. [Online]. Available: https://doi.org/10.1007/BF02289263

[16] C. Goutte, P. Toft, E. Rostrup, F. . Nielsen, and L. K. Hansen, "On clustering fmri time series," *NeuroImage*, vol. 9, no. 3, pp. 298 – 310, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811998903913

[17] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377042787901257