Protocol analyzer: Streamlining communication troubleshooting

by Digital Grid Product Department

Executive summary

Utilities are rapidly growing and modernizing their equipment, making it harder for SCADA engineers to troubleshoot communication issues. Traditional approaches are becoming unusable due to security constraints and the complexity of the entire process. EcoStruxure™ ADMS streamlines the entire process with embedded tools in just a few clicks. The ability to save the captured traffic makes it useful not just for diagnosing of issues but also for various learning purposes. This paper will delve into the capabilities of the Protocol Analyzer and demonstrate its advantages over the traditional approach of troubleshooting communication issues.

Introduction

Infrastructure systems run by electric utilities manage electrical grids and provide stable and reliable supply of electrical power to the consumers. Because of that, they have always been considered critical and for a good reason. Nowadays, with the electricity consumption at its highest, their importance is even more prominent. EcoStruxure™ ADMS is Schneider Electric's software suite for managing distribution networks with a variety of advanced functions for optimizing the usage of resources and keeping the reliability indexes high.

As a direct link between the ADMS and field devices, SCADA is one of the key components in the entire EcoStruxure™ ADMS suite. Using its strong communication protocols portfolio, it communicates with the remote entities such as Remote Terminal Units (RTUs), Intelligent Electronic Devices (IED) and external SCADA systems and ensures that the system operators always have a good representation of the state of the grid. The communication is performed through scalable Front-End Processor (FEP) in order to support configurations with large number of remote devices. After acquiring fresh values from the field, the FEP then translates the messages from protocol specific formats to unified EcoStruxure™ ADMS format, performs various checks and conversions and prepares the data for the advanced processing.

There are several industry standard protocols such as DNP3, IEC 104 and ICCP which are widely used and proven to work for most of the use cases. Still, devices that use those industry standard protocols do not all behave in the same way. While the standards define all the data types and message types that can be used, it's up to the device vendor to choose which parts of the standard to implement for a specific device type.

Although the standard protocols are mature and in existence since 1990s, many utilities still work with older devices which use custom protocols that were tailored to their specific needs. Current industry trends are leading towards the modernization of utilities' equipment enabling them to use one of the industry standard protocols, but the change cannot come overnight, and there's still a need for SCADA systems to support those older, proprietary protocols along with the standard ones.

A typical utility today owns different devices from different vendors using different protocols (standard and proprietary) and each device type has its own specific protocol profile. So, when a communication issues arise, it can be a nightmare for a SCADA engineer to troubleshoot it.

The old way

When the data stops coming into EcoStruxure™ ADMS, or when suspicious data is coming, the first thing on a SCADA engineer's to-do list is to check the data coming into the system from the field device. In the legacy systems, usually the only way to do this was to have a 3rd party tool for capturing network traffic installed on SCADA servers and to permit SCADA engineers to log on to those systems and run the tool to capture and analyze the real-time traffic between SCADA and the field device. In a happy-path scenario, when device uses one of the industry standard protocols supported by the 3rd party tool, the data would be shown in human readable format in the Wireshark, but the SCADA engineer would still have to manually map the addresses, ports and indexes at the protocol level onto the ones in the system configuration. Each manual step introduces a risk of a human error, which could make the situation even worse by prolonging the diagnosis of the issue.

In a not-so-happy-path scenario, the devices would use a proprietary protocol that is not natively supported by the 3rd party tool, and the utility would either have to implement a plugin for that tool that parses the protocol in hand, or SCADA operators would have to manually parse the raw binary data into human readable protocol messages and then to map it onto the configured devices.

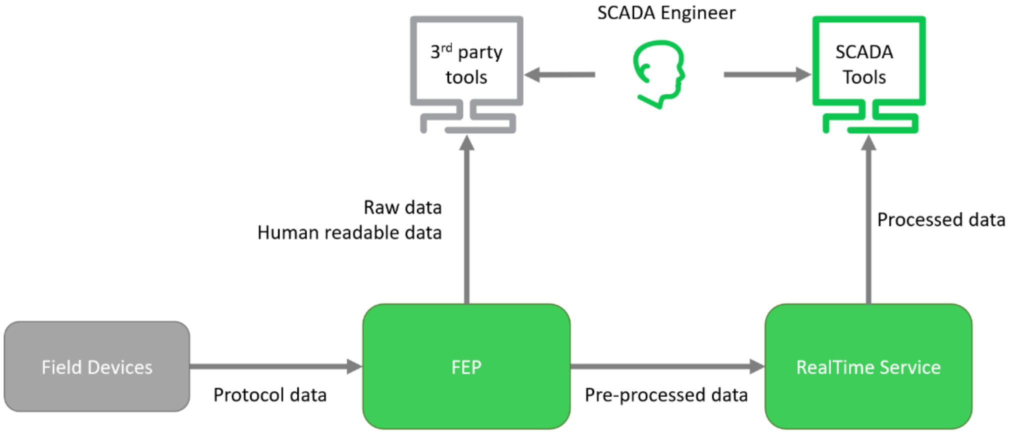


Figure 1

Troubleshooting communication issues in the legacy systems where 3rd party tools are used

Both of these scenarios assume that the existing utility security policies still allow for the 3rd party tools to be installed on the production servers. More and more utilities are tightening their security policies and having 3rd party tools on production servers is strictly prohibited. In those cases, SCADA engineers need to be very creative in order to get to the data being exchanged between SCADA and the field devices in such environments. It could even mean involving a utility's IT department to provide the needed data.

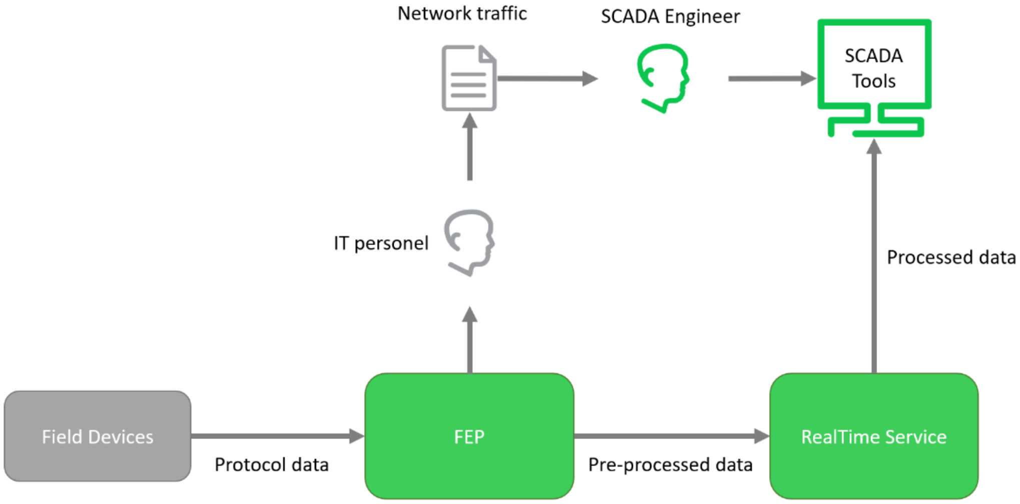


Figure 2

Troubleshooting communication issues in the legacy systems where 3rd party tools cannot be used

As it can be seen, all these scenarios require multiple manual steps and in some extreme cases even engagement of IT personnel to collaborate with the SCADA engineers to get to the needed information. Every manual step increases the probability of human errors, unnecessarily generating wasted time for tracking those errors and resolving them. In the long run, manual actions affect the total cost of ownership and weaken the system's reliability.

The Schneider Electric way

With the development of the Next-Gen SCADA within EcoStruxure™ ADMS (available from version 3.8, released in 2019) came the opportunity to address this issue. As a result of those efforts, the Protocol Analyzer was developed as an off-the-shelf tool for analysis of the protocol traffic between the EcoStruxure™ ADMS and the field devices. Having it fully embedded in the DMD (an application that SCADA engineers use in their everyday activities), a single pane of glass concept was preserved with the look and feel completely aligned with other modules and applications. Another big advantage of this approach is the fact that all troubleshooting is done from the workstation and all necessary information for decision making is accessible from a single application, so there's no need to directly access the production servers to obtain data.

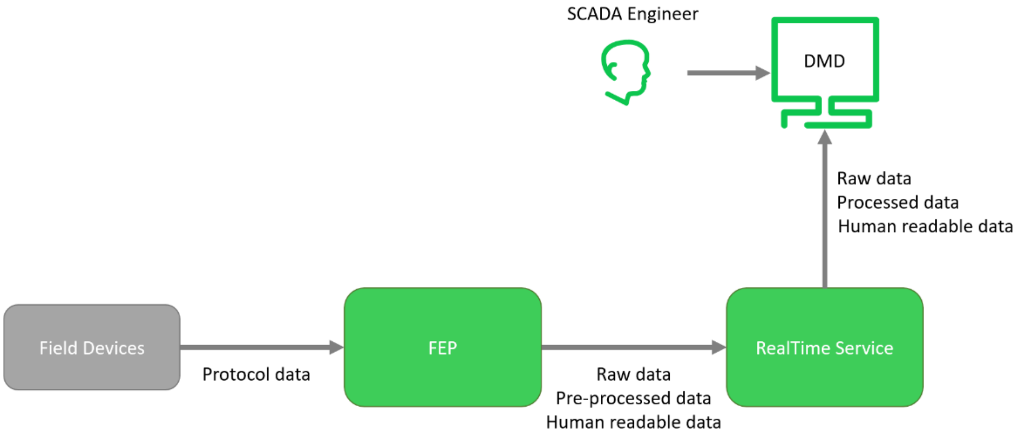


Figure 3

Troubleshooting communication issues in the EcoStruxure™ ADMS

The Protocol Analyzer supports all protocols that are implemented in the EcoStruxure™ ADMS as listed below:

- DNP3 (Client & Server
- IEC 104
- IEC 101 (Client & Server
- ICCP (Client & Server
- IEC 61850
- Modbus
- Series 5
- Vancomm
- Conitel 2025/Baker
- MD3
- PG&E 2179

Given the protocol-portfolio extensibility and the fact that the Protocol Analyzer is embedded into the EcoStruxure™ ADMS, every new protocol implemented in the EcoStruxure™ ADMS comes with the support for troubleshooting communications in the Protocol Analyzer.

For multi-layered protocols (e.g., data layer, link layer, application layer etc.), messages from all layers are supported and shown in the Protocol Analyzer.

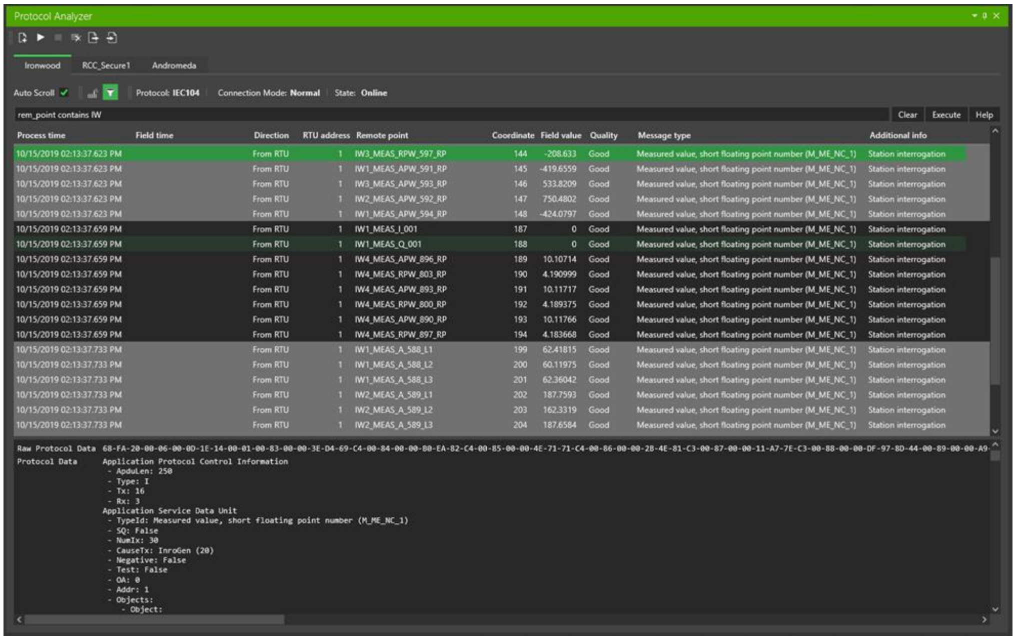
Even when the communication is performed through an encrypted communication link (e.g., ICCP over TLS), the Protocol Analyzer can parse and show ICCP messages in a human readable format, otherwise impossible with 3rd party tools. With the previous solutions, there was no easy way to see even the simplest errors during a protocol-defined handshake. However, the new EcoStruxure™ ADMS capabilities equip utilities to combine the best of the two worlds: the security provided by the TLS and the ease of troubleshooting enabled with the Protocol Analyzer.

The traffic between the EcoStruxure™ ADMS and the field devices is not constantly monitored. It can be turned on/off per Remote unit/Terminal server from the DMD so that it doesn't put any additional strain on the resources when not needed. When it is turned on, all protocol data for the selected device is parsed into a human readable format, paired with the signals/remote point in the model and sent from SCADA servers to the client application where it's presented to the user in both a human readable format and as raw data in a hexadecimal format.

Typical communication issue diagnostics process

Communication issues become visible when a certain alarm is reported to a SCADA engineer. It can be a Telemetry Failure alarm (reported when periodic scanning is constantly failing), Communication Lost alarm (reported when the communication link between RTU and EcoStruxure™ ADMS is broken), or some other type. There are numerous indicators of communication issues, but these two are the most common ones. They are usually of high importance and need inspection as soon as possible, since they are indicators that the system is no longer receiving fresh values from the field and that the operators are working with an outdated (and quite possibly incorrect) state of the grid which can impair safety.

Thanks to different navigational shortcuts in the EcoStruxure™ ADMS, the diagnosis of a problem can be started in just a couple of clicks by activating the Protocol Analyzer. Once activated, the Protocol Analyzer shows the protocol data as it arrives to the system in near-real time. It displays both incoming messages (from RTU) and outgoing messages (to RTU) in a grid.



Auto Scroll ✓ Protocol: IEC104 Connection Mode: Normal State: Online

rem_point contains IW

Process time	Field time	Direction	RTU address	Remote point	Coordinate	Field value	Quality	Message type	Additional info
10/15/2019 02:13:37.623 PM		From RTU	1	IW3_MEAS_RPW_597_RP	144	-208.633	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.623 PM		From RTU	1	IW1_MEAS_APW_591_RP	145	-419.6559	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.623 PM		From RTU	1	IW3_MEAS_APW_593_RP	146	533.8209	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.623 PM		From RTU	1	IW2_MEAS_APW_592_RP	147	750.4802	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.623 PM		From RTU	1	IW1_MEAS_APW_594_RP	148	-424.0797	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW1_MEAS_I_001	187	0	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW1_MEAS_Q_001	188	0	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_APW_896_RP	189	10.10714	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_RPW_803_RP	190	4.190999	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_APW_893_RP	191	10.11717	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_RPW_800_RP	192	4.189375	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_APW_890_RP	193	10.11766	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.659 PM		From RTU	1	IW4_MEAS_RPW_897_RP	194	4.183668	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW1_MEAS_A_588_L1	199	62.41815	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW1_MEAS_A_588_L2	200	60.11975	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW1_MEAS_A_588_L3	201	62.36042	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW2_MEAS_A_589_L1	202	187.7593	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW2_MEAS_A_589_L2	203	162.3319	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation
10/15/2019 02:13:37.733 PM		From RTU	1	IW2_MEAS_A_589_L3	204	187.6584	Good	Measured value, short floating point number (M_ME_NC_1)	Station interrogation

```
Raw Protocol Data  68-FA-20-00-06-00-0D-1E-14-00-01-00-83-00-00-3E-D4-69-C4-00-84-00-00-B0-EA-82-C4-00-85-00-00-4E-71-71-C4-00-86-00-00-2B-4E-81-C3-00-87-00-00-11-A7-7E-C3-00-88-00-00-DF-97-8D-44-00-89-00-00-A9-
Protocol Data      Application Protocol Control Information
                    - ApduLen: 250
                    - Type: I
                    - Tx: 16
                    - Rx: 3
                   Application Service Data Unit
                    - TypeId: Measured value, short floating point number (M_ME_NC_1)
                    - SQ: False
                    - NumIx: 30
                    - CauseTx: InroGen (20)
                    - Negative: False
                    - Test: False
                    - OA: 0
                    - Addr: 1
                    - Objects:
                      - Object:
```

Figure 4

Protocol Analyzer window

Efficiency gains

Protocol Analyzer reduces the time needed for the analysis of SCADA issues up to 70% compared to the old approach.

By default, the window is always showing the last received message so that the user can follow the communication without the need to scroll. From here, a SCADA engineer can get valuable answers to the following questions:

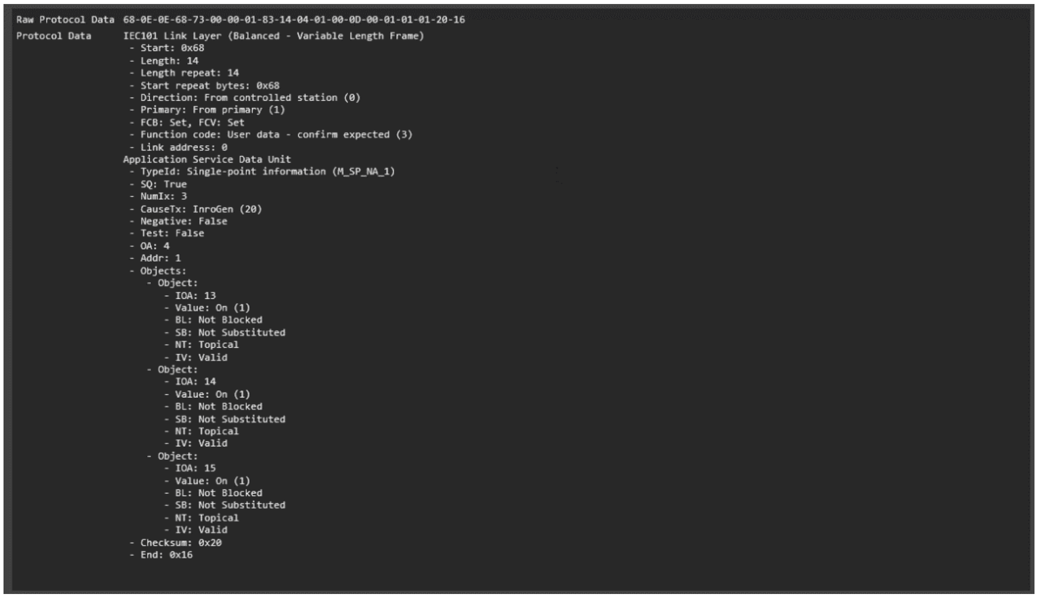
- Is the system sending the correct commands by checking the Message Type column for each sent command?
- Are the commands sent in the right order?
- Are the RTU addresses and point coordinates correct (by checking appropriate columns)?
- Are the timings OK (by comparing the RTU configuration with the actual commands being sent)?

- Is the RTU responding in an expected manner (again by checking the message type column) and in expected time?
- Is the authentication successful, where applicable (e.g. DNP3 SA, secure ICCP)?

Sometimes, just an overview of the communication in the grid can clearly tell what’s causing the issue, and SCADA engineers can take corrective measurements in a couple of minutes.

When that’s not the case, a SCADA engineer needs to dig into the details of specific messages to find the root cause of an issue. In such scenario, when a certain sequence of messages needs to be inspected in more details, the user can disable the “Auto Scroll” option in order to have the troublesome sequence in focus and inspect individual messages by clicking on them and analyzing them in the details section. This doesn’t stop the traffic capture, though, and new messages will be appended to the grid for later inspection if needed.

The details section is different for each protocol, and it shows the message structure based on the protocol definition. The data is logically grouped, following the protocol specification, and it shows all the information contained in a message in a human readable format. This can be helpful even for users not entirely familiar with the protocol specifics, since in some cases, the error indication is a part of the response, and the reason of an error is clearly visible in the details section of the response.



```
Raw Protocol Data  68-0E-0E-68-73-00-00-01-83-14-04-01-00-0D-00-01-01-01-20-16
Protocol Data      IEC101 Link Layer (Balanced - Variable Length Frame)
                    - Start: 0x68
                    - Length: 14
                    - Length repeat: 14
                    - Start repeat bytes: 0x68
                    - Direction: From controlled station (0)
                    - Primary: From primary (1)
                    - FCB: Set, FCV: Set
                    - Function code: User data - confirm expected (3)
                    - Link address: 0
                   Application Service Data Unit
                    - TypeId: Single-point information (M_SP_NA_1)
                    - SQ: True
                    - NumIx: 3
                    - CauseTx: InroGen (20)
                    - Negative: False
                    - Test: False
                    - OA: 4
                    - Addr: 1
                    - Objects:
                       - Object:
                          - IOA: 13
                          - Value: On (1)
                          - BL: Not Blocked
                          - SB: Not Substituted
                          - NT: Topical
                          - IV: Valid
                       - Object:
                          - IOA: 14
                          - Value: On (1)
                          - BL: Not Blocked
                          - SB: Not Substituted
                          - NT: Topical
                          - IV: Valid
                       - Object:
                          - IOA: 15
                          - Value: On (1)
                          - BL: Not Blocked
                          - SB: Not Substituted
                          - NT: Topical
                          - IV: Valid
                    - Checksum: 0x20
                    - End: 0x16
```

Figure 5

Details section of the Protocol Analyzer

Beside the parsed, human readable protocol data, the details section also shows the raw protocol data presented in a hexadecimal format. If a utility has its protocol simulator, this data can be used for creating test cases that can verify if a certain functionality is implemented and/or configured correctly.

Adaptive filtering for point-related issues

When an issue on a single remote point occurs, it is usually indicated by a flag on the associated signal. From the signal’s control window, the SCADA Engineer can start the troubleshooting process in two clicks.

In case of smaller RTUs (RTUs with small number of points), it is possible to see all relevant information in the grid without any further actions, since messages containing multiple remote points (such as response to a scan command) are broken into multiple grid lines so that each remote point is shown in a separate grid row. This makes troubleshooting point-related issues a little easier, since all relevant data (such as coordinate, point value and protocol qualities/flags) are visible straight from the grid on a per-point basis. However, nowadays, utilities tend to have RTUs with large number of points and finding the faulty ones in the responses can become difficult.

For those kinds of situations, the Protocol Analyzer has a built-in filtering capability used for hiding the irrelevant messages (or points) so that the user can focus only on the relative ones. In this case, the user can create filter by remote point name or a coordinate and inspect only messages related to the remote point being investigated. As in the previous case, if the data in the grid is not sufficient to discover the cause of the issue, the user can check the details section for more protocol data that can be used in analysis.

Monitoring multidrop communication

All multidrop configurations (when multiple RTUs share the same communication link) can also be observed in the Protocol Analyzer. These configurations can be challenging to troubleshoot because messages from all remotes come to the system through a single TCP/IP port.

For analyzing multidrop issues, the Protocol Analyzer can be started for the Terminal Server entity (which is hierarchical parent of the RTUs sharing the same communication link). The grid will show messages for all RTUs under that Terminal Server. From here, the user can see how actions on one RTU affect the other RTUs. The filtering capability can be helpful here as well. The user can create filters by RTU address or even combine them with the point coordinates if the issue is related to a specific point.

Protocol analyzer as a learning tool

Beside its main intended purpose, the Protocol Analyzer can also be used as a learning tool. By using it, the utilities can train their SCADA engineers and show them the structures of protocol messages on a real-life example. Since all protocol data is shown in a human readable format, this can make learning of new protocols much more efficient and much less tiresome since it doesn’t rely only on reading protocol standards.

SCADA engineers can also learn how the system is working under-the-hood and see how different combinations of parameters on an RTU can affect the behavior of the EcoStruxure™ ADMS using the Protocol Analyzer.

All captured traffic can be exported to an XML file, and reimported to any other workstation with DMD. In this way, the traffic can be analyzed offline as well, making it ideal for experienced user to capture some interesting real-life situations in the protocol analyzer and use them as training material in the onboarding process for new SCADA engineers.

Conclusion

In today's fast-paced world, there's no room for complicated and lengthy procedures, especially when working with mission-critical systems. Having cyber-security concerns as a part of those procedures only increases the necessity for a utility to advance their SCADA troubleshooting approach.

The EcoStruxure™ ADMS streamlines the troubleshooting approach by having its Protocol Analyzer available for SCADA engineers in just a couple of clicks in the same client application that is used in their daily routines. Even though it is within easy reach, it does offer many advanced features that can significantly reduce the time needed for getting to the root cause of a communication issue. This increases the system availability – a paramount for any utility and, at the same time, lowers the total cost of ownership. What's more, the Protocol Analyzer enables users to save the captured protocol traffic that can be leveraged not only for offline analyses, but for different kinds of training as well which makes it a truly multi-purpose tool.

EcoStruxure™
Innovation At Every Level

EcoStruxure™ is our open, interoperable, IoT-enabled system architecture and platform. EcoStruxure delivers enhanced value around safety, reliability, efficiency, sustainability, and connectivity for our customers. EcoStruxure leverages advancements in IoT, mobility, sensing, cloud, analytics, and cybersecurity to deliver Innovation at Every Level. This includes Connected Products, Edge Control, and Apps, Analytics & Services. EcoStruxure™ has been deployed in 480,000+ sites, with the support of 20,000+ system integrators and developers, connecting over 1.6 million assets under management through 40+ digital services. Find out more about EcoStruxure Click Here

Contact us

For feedback and comments about the content of this white paper:

ADMS Product Center Novi Sad
office@schneider-electric-dms.com

If you are a customer and have questions specific to your EcoStruxure™ ADMS project:

Contact your Schneider Electric representative at
technical.marketing@schneider-electric-dms.com