exchanges can be verified, but not the timing performance. This is because the messages do not travel over realistic distances as they should in industrial applications. By design, timing is a very important part of IEC 61850 message exchanges and needs to be investigated. While it is not possible to use cables that are several meters or kilometers long in the lab to replicate realistic situations, it is possible to insert *network emulators* in between. As shown in Figure 4, in this fashion, messages travel over long distances through networks with traffic. This ensures that timing performance of IEC 61850 messaging can be examined.

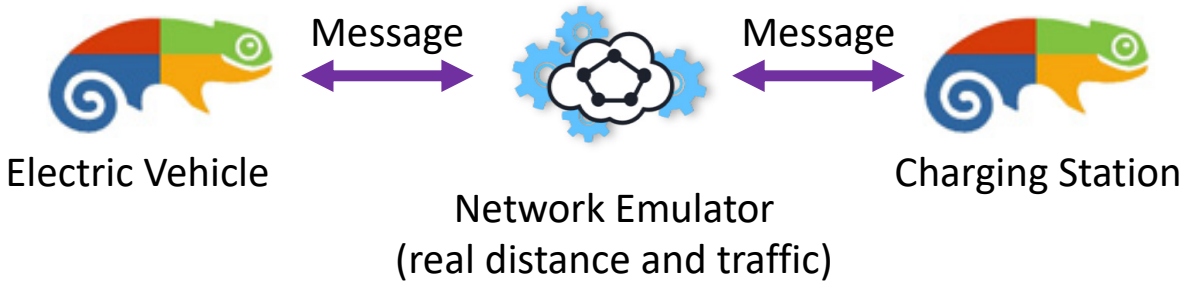

Figure 4. Realistic network and traffic design with Network Emulator.

The software tools for these three distinct steps are reviewed in the next sections. Technically, network sniffers can be classified as another category, but WireShark [44] is so dominant and works so well that it is the only tool used in the trade. It is also freely available. There simply is no reason to try any other tool. It is efficient, effective and comes at no-cost.

3. IEC 61850 Modeling and Emulator Tools

In this section, tools that can develop and run IEC 61850 device models are critically analyzed. Several factors have been taken into account such as capabilities, cost, time required for setup and customer service. The last two factors are especially important for free and paid tools, respectively.

3.1. Infotech

Infotech is a small company that provides several tools for IEC 61850 testing [45]. The software can only be run in Windows which is the biggest limitation of these tools. There are six different components that come with this package, as shown in Table 1. What is most striking about Infotech tools is their *simplicity, ease of use and cost-effectiveness*. These will be explained below.

Table 1. InfoTech IEC 61850 Software Package.

Name	Description
ICD Editor	Create and modify ICD files
61850 Avenue	Emulates Client Devices in IEC 61850 Network
SCL Runner	Emulates Server Devices in IEC 61850 Network
GOOSE Sender/Receiver	Simple interface to send/receive GOOSE messages with different variables and configuration settings
SV Sender/Receiver	Simple interface to send/receive SV (9-2LE) messages with different variables and configuration settings

ICD Editor is a strong editor tool that can be used to generate or modify ICD files. It has a simple user interface as shown in Figure 5. It gives a tree view of any device. Logical

Devices (LDs) and Logical Nodes (LNs) can be added as shown in Figures 6 and 7. It is also possible to add Control Blocks (CBs) and Data Sets (DSs) as shown in Figure 8.

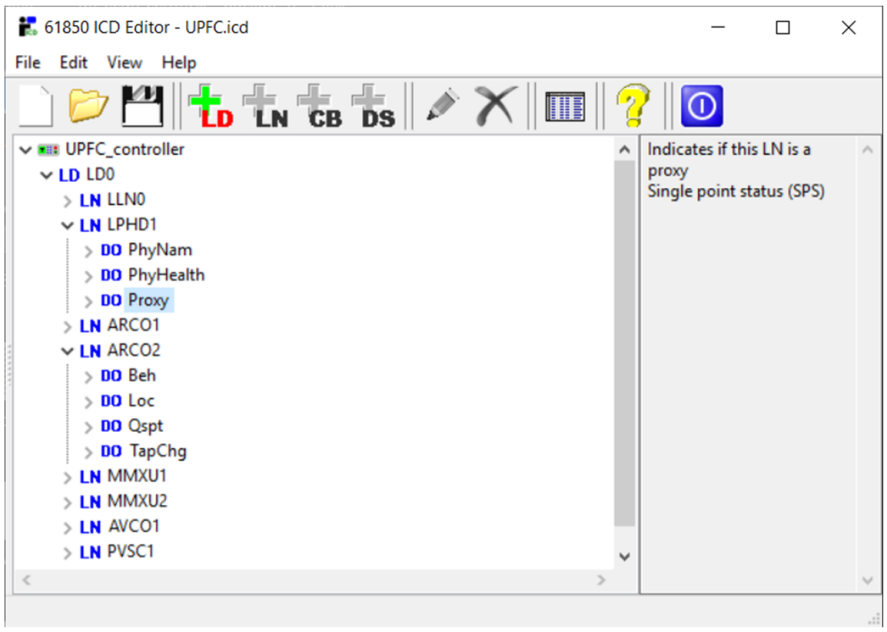

Figure 5. General view of ICD Editor.

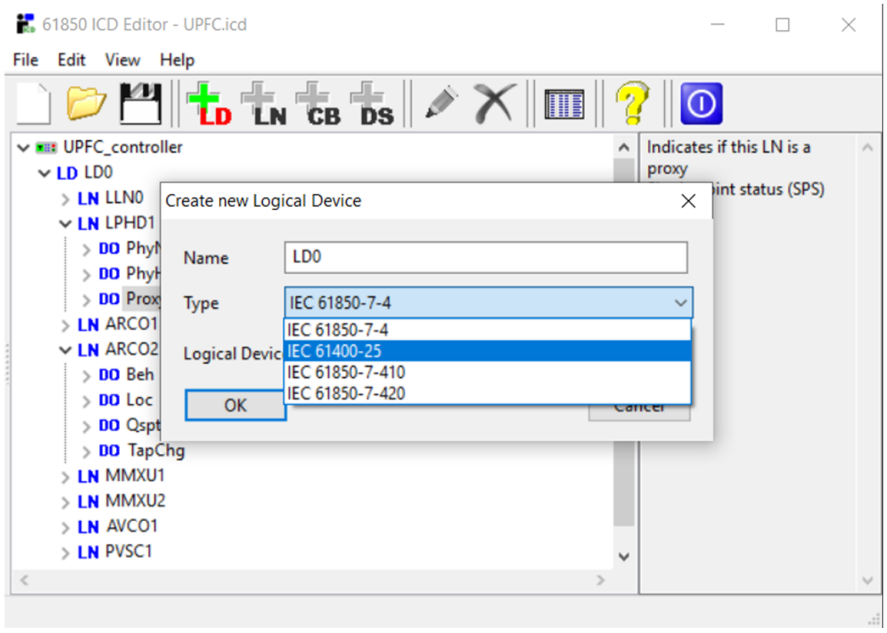

Figure 6. Adding Logical Device from IEC libraries.

The only limitation of this tool is that LDs can only be added from published IEC 61850 libraries so that they will be compliant with the standard. When developing novel IEC 61850 models for devices that are not yet included in the standard [46–48], this becomes a barrier. One possible way to work around this is by using the XML editor capability of the ICD editor tool. This shows the XML rendition of the ICD file created as shown in Figure 9. Following the XML rules, it is possible to add novel LDs and LNs into an IEC 61850 model. This approach provides all the necessary capabilities for ICD editing in the IEC 61850 research field.

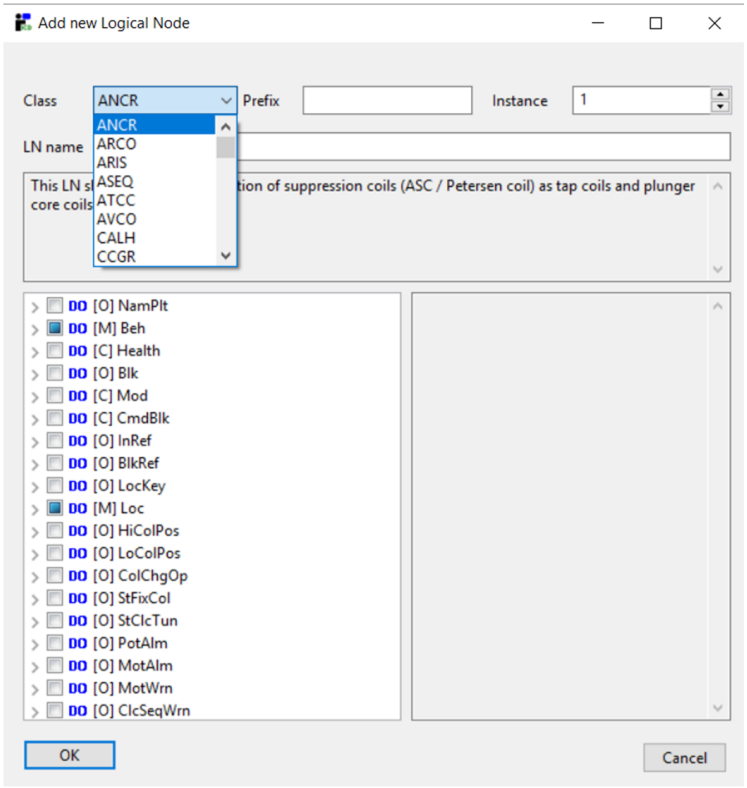

Figure 7. Adding Logical Node from IEC libraries.

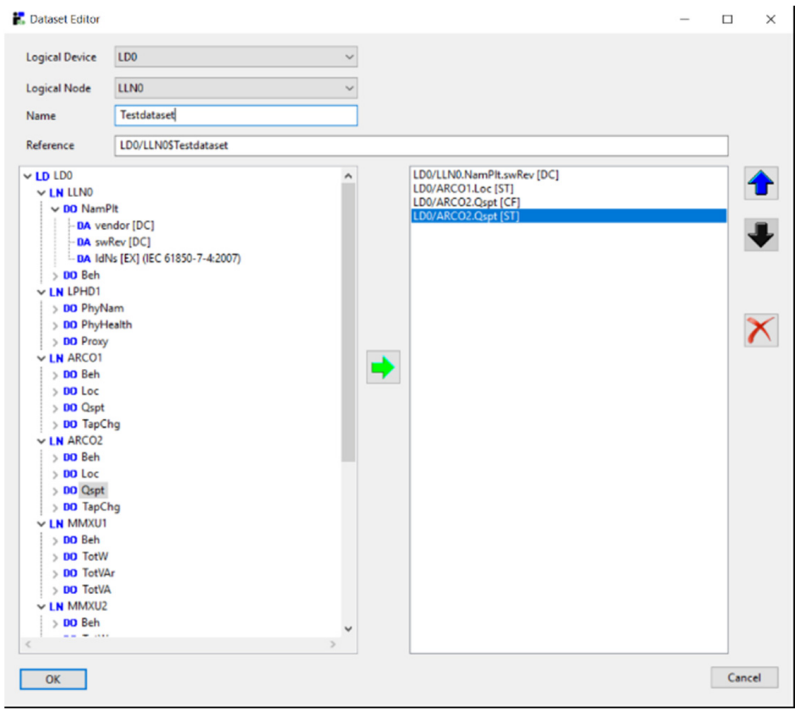

Figure 8. Creating a DS within IDS based on LDs and LNs.

Once necessary ICD files are prepared, 61850 Avenue and SCL runner tools are utilized to emulate client and server devices, respectively. Proper connection can be established, local data changes can be reported to the remote terminals and necessary instructions can be sent. All the actions can be observed in real-time and messages can be captured in the

network. Figure 10 shows how one or more clients are emulated in 61850 Avenue. One device (laptop or pc) can emulate several clients but only a single server.

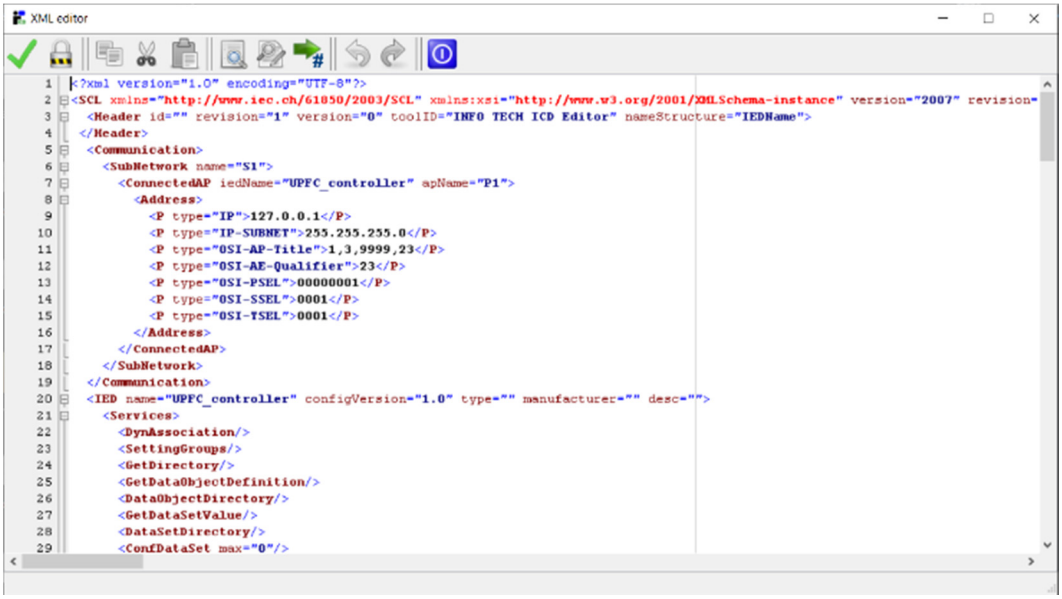

Figure 9. XML editor available in ICD Editor tool.

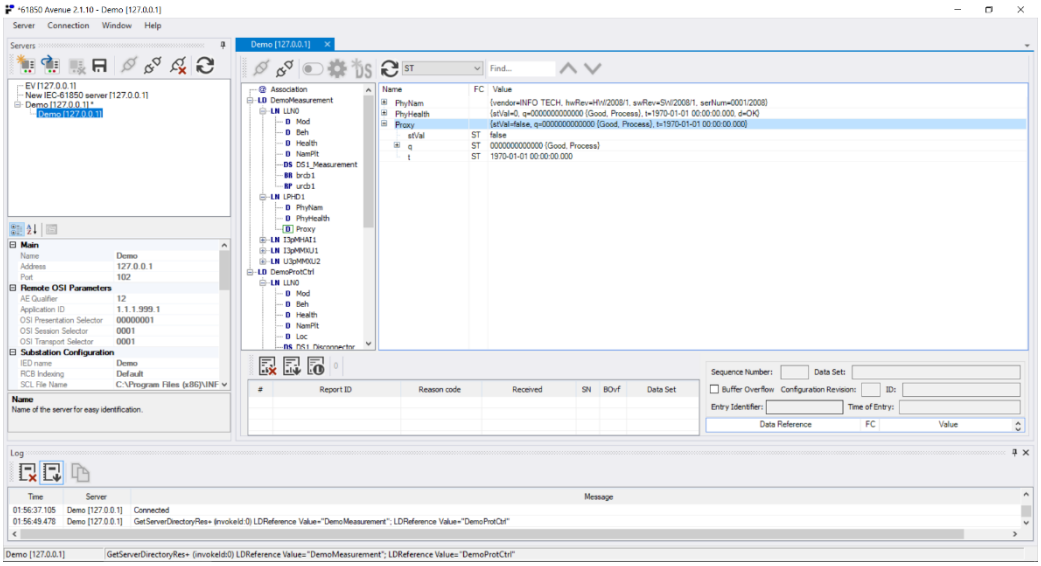

Figure 10. Avenue Interface with Server Information.

It is possible to add multiple servers on the left to which Avenue should connect as a client. With the recent update, this connection can use Transport Layer Security (TLS) with certificate authentication, and a private key. Once the connection is established, server device data model can be explored, and the current data values can be read. If there is a change in the parameter values this is denoted with blue ink.

As shown in Figure 11, it is possible to create new data sets from the server's data objects. It is also possible to set up reporting conditions based on data sets defined in the server's ICD file or created in Avenue. It goes without saying that it is possible to send control commands as real or test messages as shown in Figure 12.

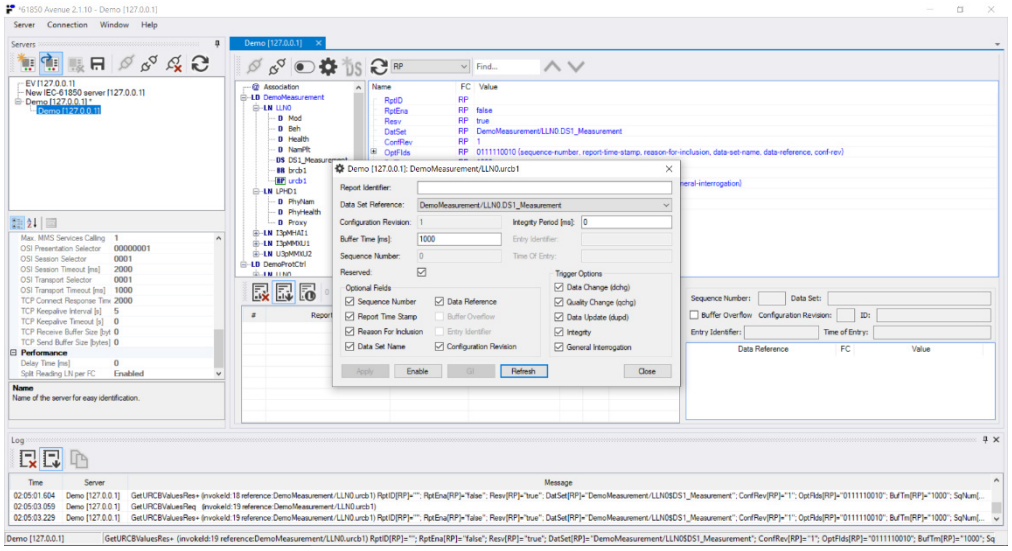

Figure 11. Reporting in Avenue.

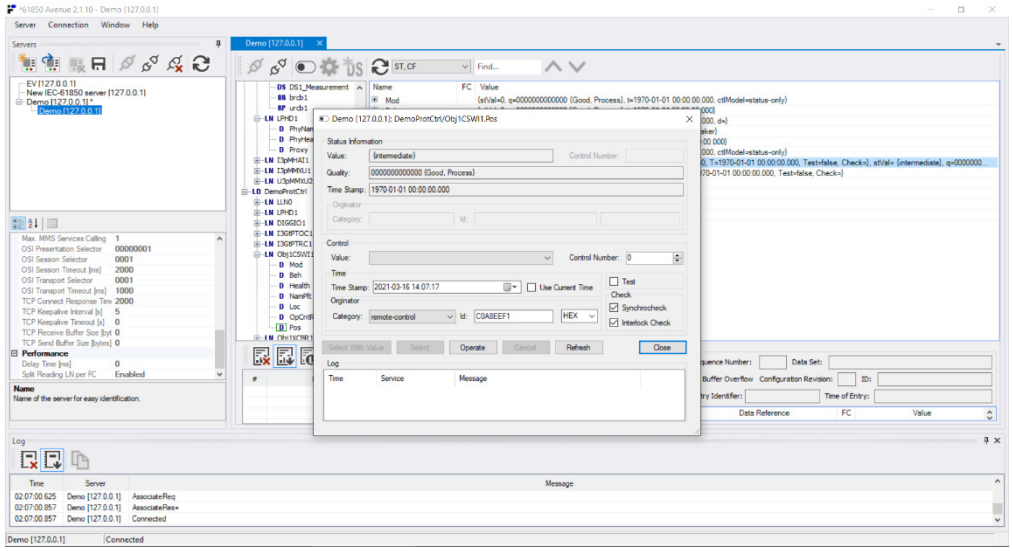

Figure 12. Control Commands sent as real or test messages.

On the server side, SCL Runner is running the ICD file of the server and emulating it. As shown in Figure 13, several servers can be added with their ICD files and IP configurations. At any given time, only one can be launched, and the computer emulates that particular device.

Avenue and the SCL runner can be run on different platforms or as a loopback setup. In either case, when the client and server are connected, the parameters can be edited, and changes can be observed on the opposite side. If the server makes the changes locally, client receives updated information in blue ink, as shown in Figure 11. On the other hand, client can ask for a parameter to be changed in the server, which is reflected inside the server if this modification is allowed.

GOOSE and SV messages can be triggered from Avenue and SCL runner tools as long as the message configurations are done within ICD files, such as goose control block GOCB. However, sometimes researchers would like to focus on the message design and their contents, more than on full-fledged ICD file development. For such cases, there are simple sender and receiver tools for both GOOSE and SV messages. As shown in Figure 14, GOOSE sender has a simple interface where network parameters of the message can be easily configured. Furthermore, GOOSE block data set can be created by adding parameters

from a drop-down menu. When the transmission starts, the tool starts sending GOOSE messages through the chosen network adapter. Initially, only simple GOOSE messages could be sent. However, with the recent update, routable-GOOSE, R-GOOSE, can also be sent with tunneling options.

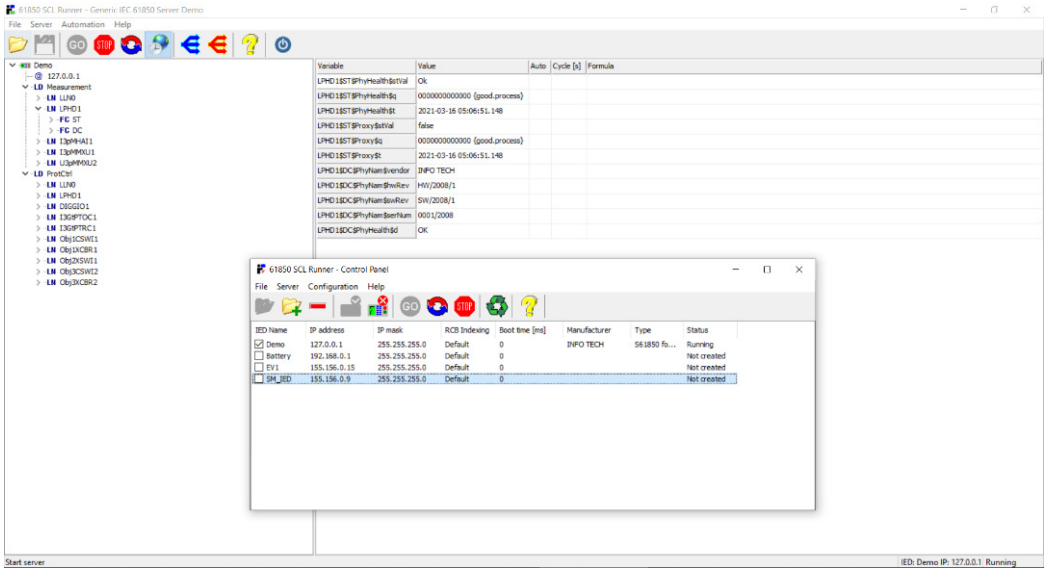

Figure 13. SCL runner emulating a server.

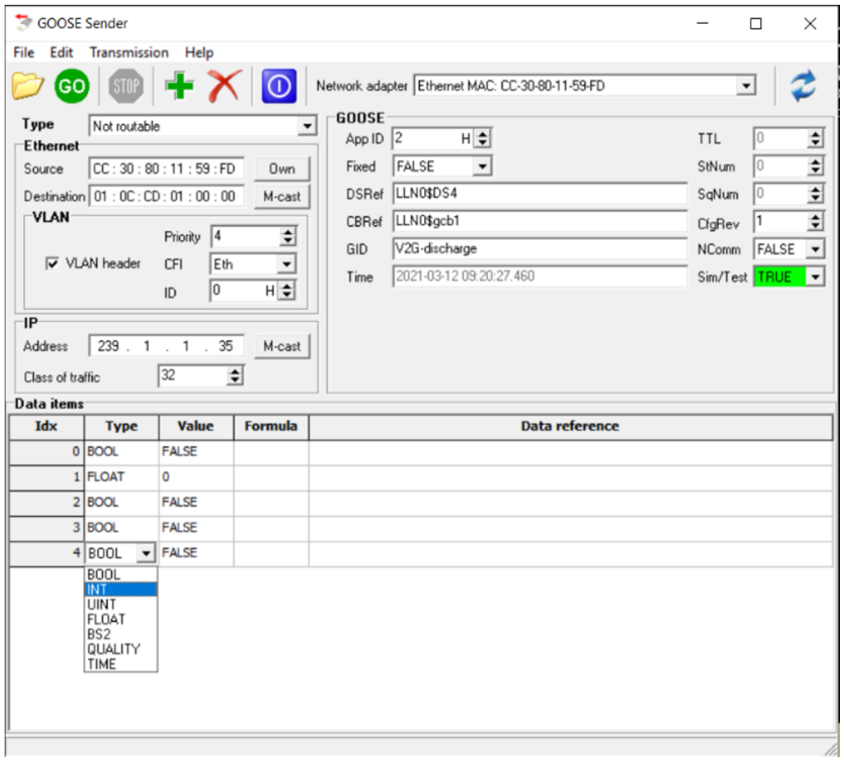

Figure 14. GOOSE Sender Interface.

The receiver tool's interface, as shown in Figure 15, is almost a mirror image of the sender. The only difference is that everything is read and shown, not editable. The receiver can detect available streams in the network, depending on the selected adapter. Once subscribed, the contents of the GOOSE message are shown in a user-friendly manner. As

shown in Figure 16, the tool also has a parser window for viewing more technical details of the message.

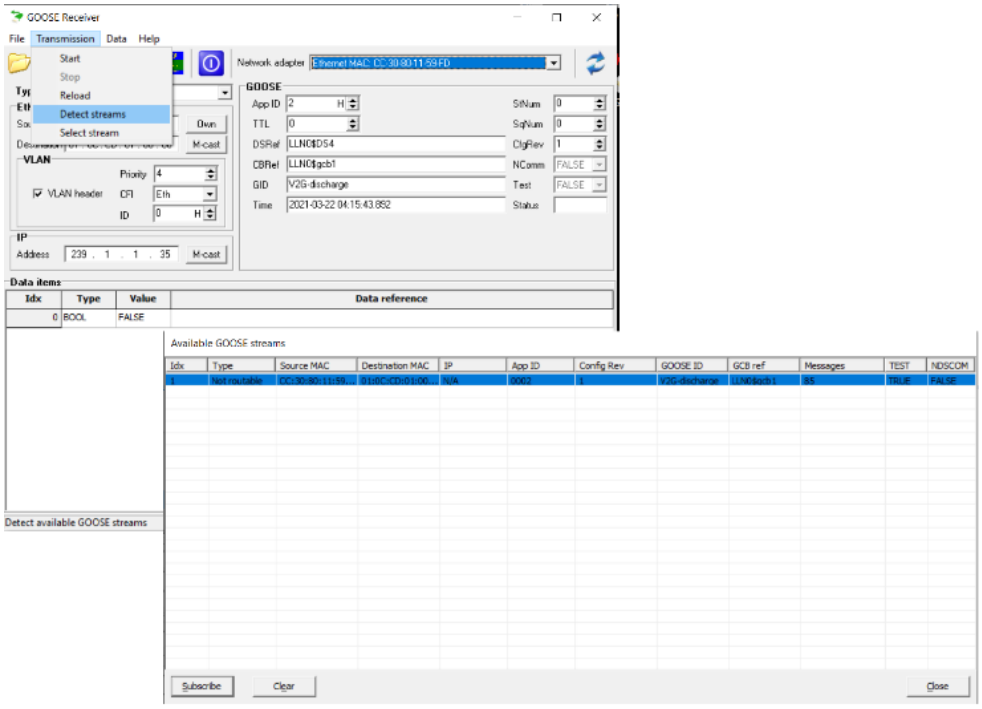

Figure 15. GOOSE Receiver Tool and Stream Detection.

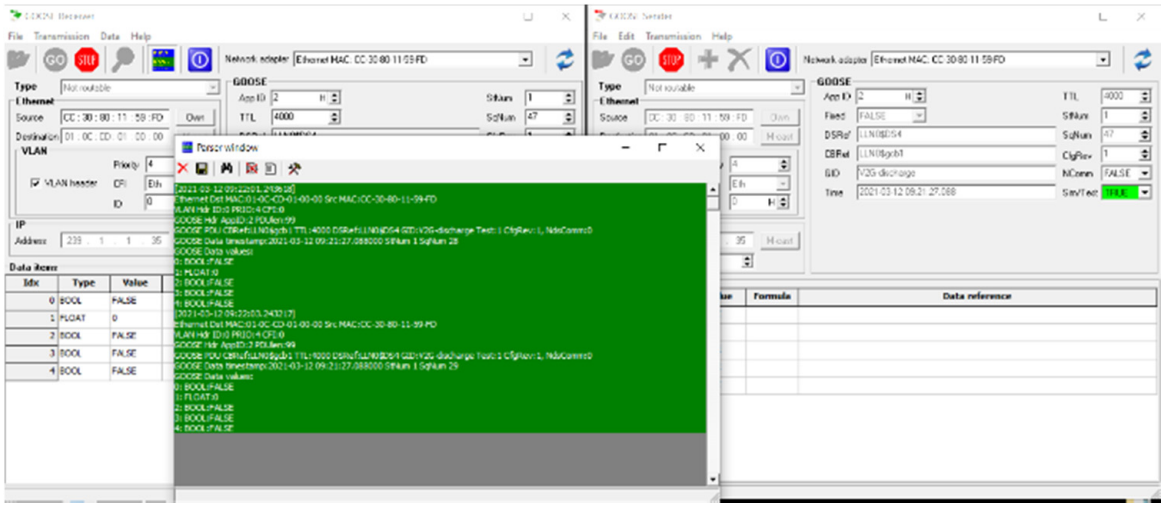

Figure 16. Message Parsing in GOOSE Receiver Tool.

These sender and receiver tools need not be used as a pair. They are interoperable with any valid GOOSE message. Therefore, they offer a very convenient method of testing when the main focus is sending and receipt of messages.

SV Sender and receiver tools, (see Figures 17 and 18), operate in almost the same way, with a significant difference. Industry has agreed on a very limited implementation of SV messages that is called 9-2LE, i.e., limited edition [49]. This is due to the fact that, in their general definition, SV messages are very general, and industrial partners found it hard to implement. Following this trend, SV sender and receiver tools can only deal with 9-2LE messages. For industrial purposes, this ensures capability. However, as in ICD editor tool, for cutting-edge research it creates a handicap. Novel SV message designs and their

operations cannot be implemented or tested, e.g., use of SV for much slower messages such as EV charging signal [50].

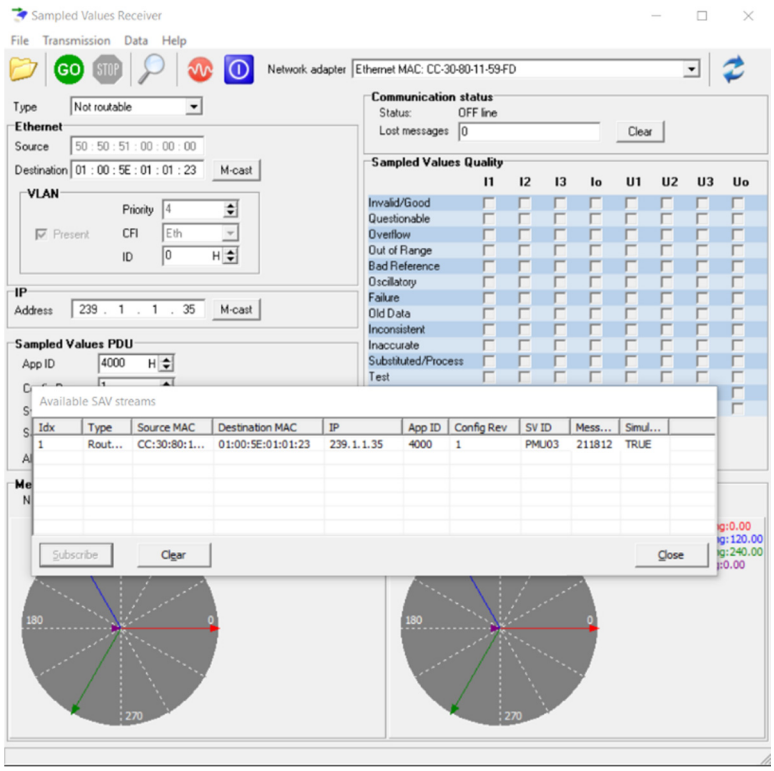

Figure 17. SV Sender Interface.

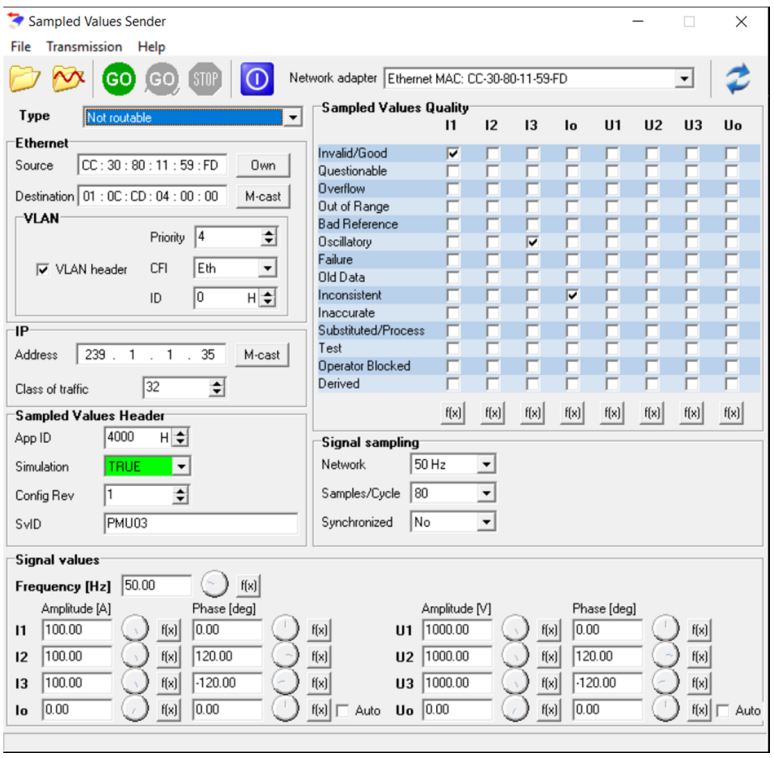

Figure 18. SV Receiver Detecting 9-2LE streams.

In addition to technical properties, this work evaluates these tools in terms of cost, customer service, experience with the company, etc. The overview of Infotech tools evalua-

tion is given in Table 2. From this perspective, it can be easily said that Infotech passed all these fields with flying colors. Firstly, the cost of the entire suite is very reasonable. The company is very responsive and communicates well. They acknowledge the receipt of payment, and then follow up with product delivery, licensing and shipping of physical keys as shown in Figure 19. The only negative aspect of Infotech products is that these keys are the purchased value. In other words, if these USB flash keys are lost or became unresponsive, the customer would bear the cost of another purchase. That being said, dongles have an advantage, where the tools can be installed in a number of computers and keys can be carried with ease, instead of carrying the computers around. This definitely has advantages along with drawbacks.

Table 2. Infotech Tools Review Summary.

Evaluation Criterion	Score	Notes
Tool Capabilities	5—Excellent	SV sender and receiver tools only support (9-2LE) ICD Editor only supports standardized LNs, LDs. XML editor should be used for novel LN/LD designs
Cost	5—Great Value for Money	Infotech costs 10 % of its competitors but does the same job in a reliable and better way
Customer Service	4—Responsive and Active	Online Training is available immediately after purchase, Updates are sent by the company in a proactive way
Licensing	3—Convenient	USB flash keys are used for licensing. Malfunctioning or loss is always a possibility. It also makes the use of several platforms easier.
Time required for Installation and Use	5—Very Robust and User friendly	Installation packages are reliable. Installation takes less than 15 min. Any person with basic IEC 61850 knowledge can master the tools in a day. Online training is also included with the purchase.
Installation is very smooth.	5—Excellent	The tools are robust and reliable. Even after long periods of idle time, they work as expected with no issues. Interoperability tests with other tools also proved their successful operation.

In terms of customer service, Infotech is very responsive and helpful. The company set up online training sessions within one week of purchase. The trainers extensively discussed the capabilities and use of these tools. Most importantly, the use of the tools is intuitive, and they are reliable. The menus, icons and menus are very easy to master and remember. Once this is done, it is convenient to use these tools daily, or only once in a while. Sometimes, research projects go cold, and researchers want to revive them. Infotech tools proved to be easy to remember and reliable to work even after months of non-use. Despite being taken for granted, these qualities are not always present, as will be seen in the next tool.

3.2. Xelas Energy

Xelas Energy software is a scavenged tool from old communication emulation programs. Bootstrap solutions are developed to emulate IEC 61850 servers and exchange IEC 61850 messages. It is also claimed that IEC 62351 cybersecurity requirements are built into this tool. However, it is not possible to over emphasize that Xelas Energy is reuse of an obsolete software. It is not a freshly developed tool and this manifests itself in all aspects such as poor performance, little to no reliability and very low user-friendly interface.

Surprisingly, these are not the worst aspect of Xelas Energy, but customer experience and interaction with the company. Firstly, technical capabilities will be presented.

Figure 19. Infotech Flash Keys.

This tool is provided as a collection of different packages in one installation file. Contents are not clearly listed and there are many dependencies. As shown in Figure 20, the tool is not self-contained but is a collection of different things. The GUI is provided by an Internet Browser, as the tool does not have its own GUI. Energy Management System is not relevant to anything, although it is always mentioned in documentation along with the tool itself (see Figure 21). Its operation depends on the Glassfish [51] server which, then, communicates with IEC 61850 stack to exchange messages. This requires *tp0d* to be utilized. Due to bad design of Xelas Software tool, *tp0d* is not only occupied when the tool runs, but also when it is closed. This means that other software that require access to *tp0d* cannot be run on the same computer. Similarly, glassfish is run in each boot cycle, takes up memory and slows down the computer.

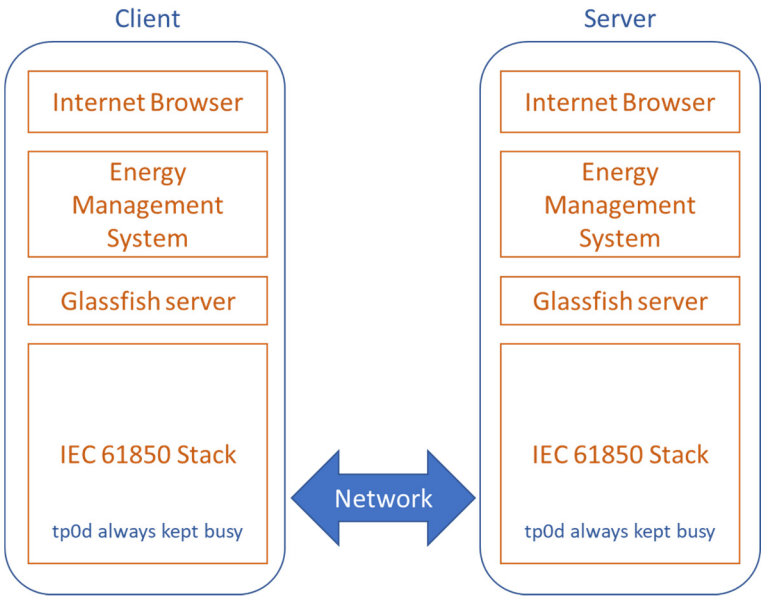

Figure 20. Xelas Energy Architecture.