3D at Depth

3D at Depth, Inc

SL Sensor

Real Time Viewer

TCP Communication

Manual

Version C4

November 22, 2021

Change Record

Version	Date	Notes	Author
C.1	8 May 2019	Initial Release	J. Jeong
C.2	17 Feb 2020	Replay of Data	D. Florin
C.3	7 Jul 2021	Added CCI Test usage, updated images	D. Florin
C.4	22 Nov 2021	Cleanup of Text, table update, line scan size is no longer only 500 lines.	D. Florin

Table of Contents

1 Definition and Acronyms

ROV: Remote Operating Vehicle

CCI: Command and Control Interface

RTV: Real Time Viewer is a software program which shows Lidar scanning data in real time. This normally is located on the control laptop.

EC: Embedded Controller is the software which receives commands and controls the sensor. This is located on the SL sensor.

InstaCloud: InstaCloud is a software which processes the raw Lidar data and sends it to RTV and is located on the SL sensor.

3D Collect: 3D Collect is a user-friendly GUI (Graphic User Interface) to help command the sensor. This is normally located on the control laptop.

SL Sensor: Subsea Lidar Sensor

Riaat: Range, Intensity, Angle, Angle, and Time data in a 3D at Depth defined format

UDP: User Datagram Protocol

CCI Test: Application developed to test the communication between RTV and a Client, both sides. Allows replay of data.

2 Introduction

This document explains how to communicate with the Real Time Viewer through TCP/IP protocol. The goal is for the 3D at Depth system to provide 3D data to the CCI (Command and Control Interface) package to enable “real time” mapping in global coordinates.

The detailed Riaat-CCI packet description is in a separate document 3DatDepth_CCI_packets_RevC7.

3 Overview

This manual describes how the 3D at Depth Lidar system works with the CCI software for the system.

The SL Sensor has two programs, EC and InstaCloud. EC receives commands, controls the SL sensor, acquires the data, transfers the data to InstaCloud, and reports its status. InstaCloud converts the raw data to Riaat format, may store Raw and/or Riaat data on the sensor, and transfers the Riaat data to the Real Time Viewer.

The Real Time Viewer opens a TCP port as a server for external communications. CCI can connect to RTV with the IP address of the RTV. The port number for the server is 2019.

Riaat-CCI type packets will be transferred through the TCP connection. The structure of Riaat-CCI packets is described in a separate document 3DatDepth_CCI_packets_RevC7.

Figure 1 below is a schematic of the overall communication plan for Sabertooth system. Green boxes in the Figure are 3D Depth software. In this document, we focus on the Real Time Viewer.

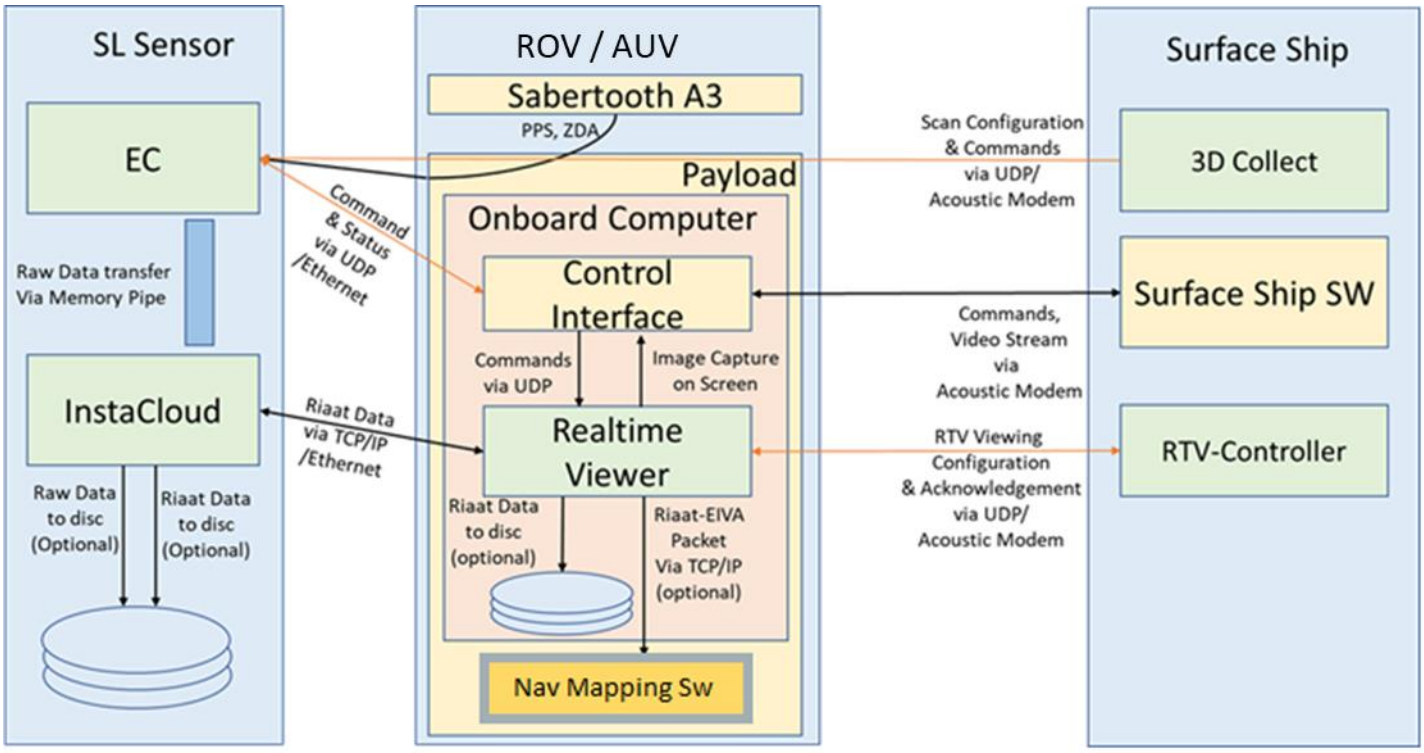


Figure 1. Overall Communication Schematic

4 Test System settings

For testing, a SL2 sensor was connected to the simulated Sabertooth computer (top and middle in Figure 2) through a switching hub, and the simulated topside commanding center (top and left laptop in Figure 2) was also connected to the same switching hub. The CCI simulator was also connected to the same switching hub. The CCI simulator was built solely to validate successful data transfer.

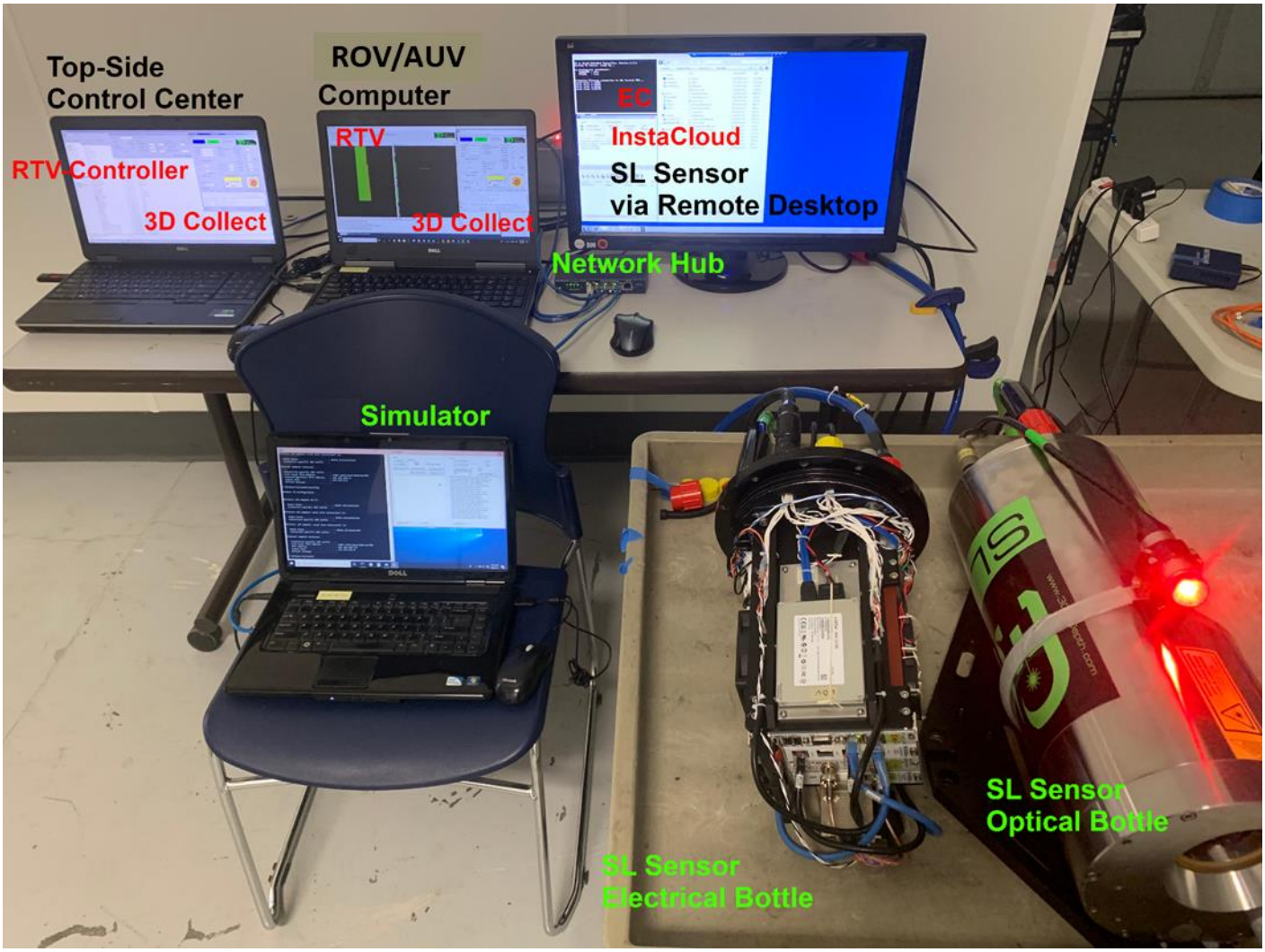


Figure 2 Test System Overview

The top side control center runs 3D collect and ICD-RTV controller. The ROV/AUV computer runs Real Time Viewer, and also opens a remote desktop to the SL sensor. The CCI simulator program runs on a different computer to connect to RTV via TCP.

4.1 Real Time Viewer

While scanning is in progress, the RTV receives the riaat data from InstaCloud and displays the 3D data on the screen in “real time”. The Real Time Viewer also opens a TCP port for external connection when the ‘Open Server for CCI Connection’ checkbox is on. The checkbox also enables the CCI communication Status LED on the panel. When connected, the LED turns to green as seen in Figure 3.

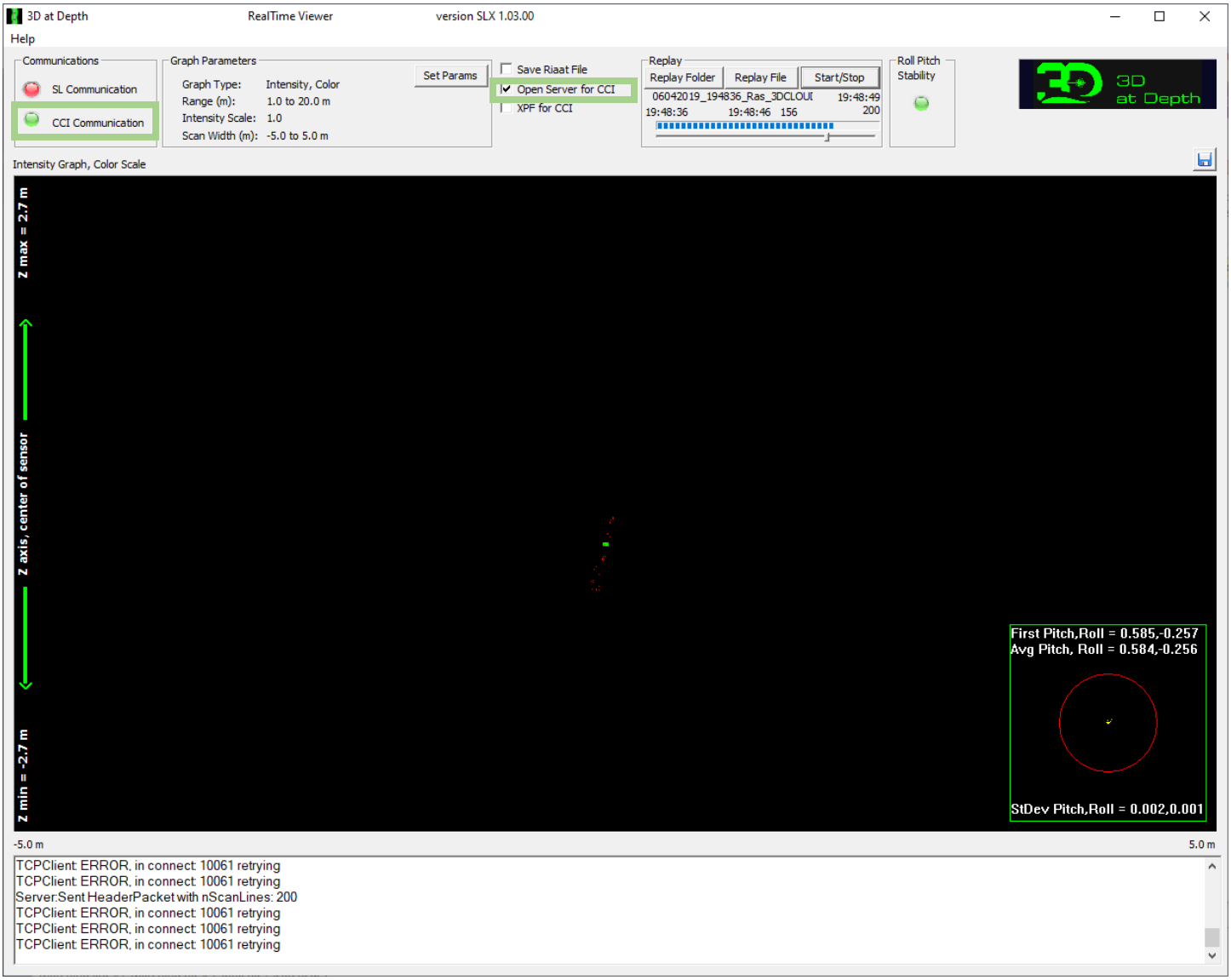


Figure 3 Real Time Viewer

RTV sends a header packet of scanning conditions and scan Line data to CCI. For a ‘Raster Scan’, one header packet is sent before scan line packets are transferred. Each scanline is packed in a packet. Detailed information on packets are described in a separate document 3DatDepth_CCI_packets_RevC7.

The Real Time Viewer in ‘LineScan’ mode, saves a riaat file for every 500 scanlines (although this is going to be more configurable in future versions of the code and could be either larger or smaller). RTV sends Scan Line Data Packets to CCI for every scan line received from InstaCloud. RTV also sends the header packet repeatedly for every 500 scanlines (or whatever the configuration defines) while scanning is in progress.

The Real Time Viewer may drop some scanlines to the CCI when the RTV TCP communication buffer is full. The system receiving the riaat-CCI packets, which is the CCI simulator in Figure 2, may not be fast enough to consume the transferred data. This results in the receiving TCP buffer filling up, which can then cause the RTV buffer to fill. In this scenario, the RTV needs to drop some scanlines since the data from InstaCloud comes steadily in, but the output buffer is full.

4.2 Replay of Data

The RTV offers the ability to replay a scan back. This involves loading a RIAAT file and looking at the timestamp on the data and pushing the data into the receive buffer so that it follows the same data path as the original data. This includes sending the data onto the connected interfaces in the expected format. To further assist in testing of software, the replay will also find any applicable XPF, or ASCII, navigation data file and open an interface that can deliver the iXBlue defined Navigation Long packet that has Inertial Navigation Data so that the RIAAT data can be placed into true three-dimensional space in world coordinates. The Long packet data is time correlated with the RIAAT as the original one would be. This does not have to be used if the receiving system has another way to pull in the navigation data for the scan data.

While sending Navigation Long packet functionality is in the RTV, the only users of the INS data so far has been the simulated interface and as such it has not been fully, independently, validated. This must be considered experimental code until the long packets can be processed properly and independently validated.

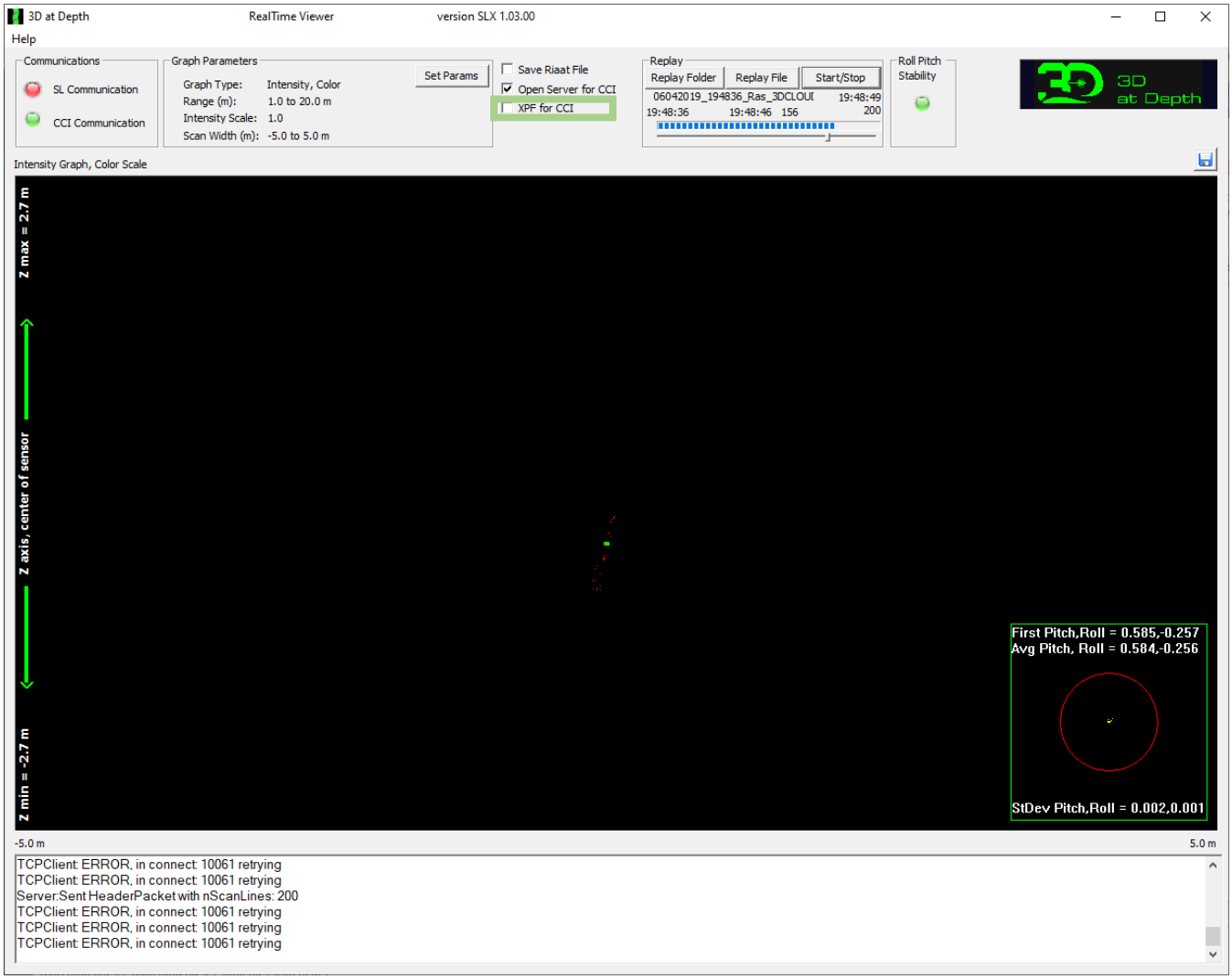


Figure 4 Real Time Viewer, XPF for CCI

4.3 CCI Test Application

The CCI Test application offers a way to test both the RTV and a Client Mapping application that wants to receive data from the 3D at Depth system. This application stands in for the RTV and allows sending data at a much slower rate as well as testing things like error packets. The application is a development tool to make it easier to integrate with the RTV product.

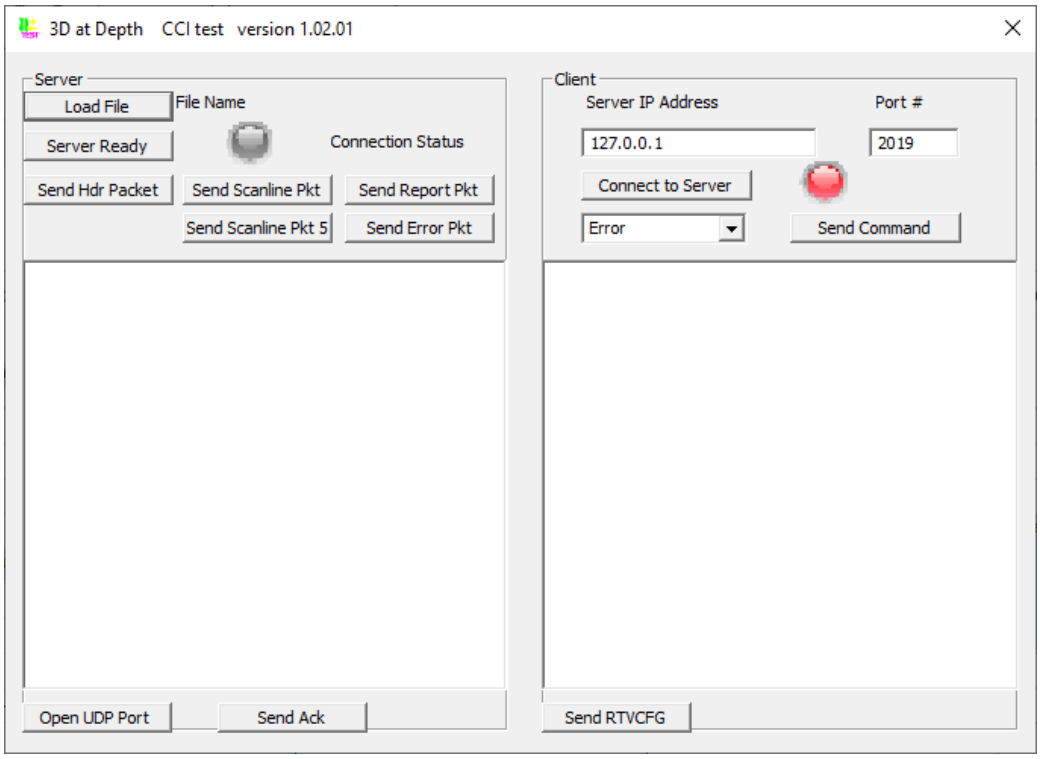


Figure 5 CCI Test

4.3.1 Client Side

The Client side (right side) is the part that will attach to the RTV and is used to simulate a Client Mapper that consumes the CCI packets sent. The **Server IP Address** is 127.0.0.1 if the application is running on the same computer that is running the RTV application. If the application is running on a different computer, then the IP address of the RTV computer is needed (and the network connection needs to be IPV4 based and easiest if it is local, otherwise the packet may get stripped out by routing). The IP address and port has to be allowed through firewalls. Test with using a command prompt and executing the command:

xxx>ping 127.0.0.1

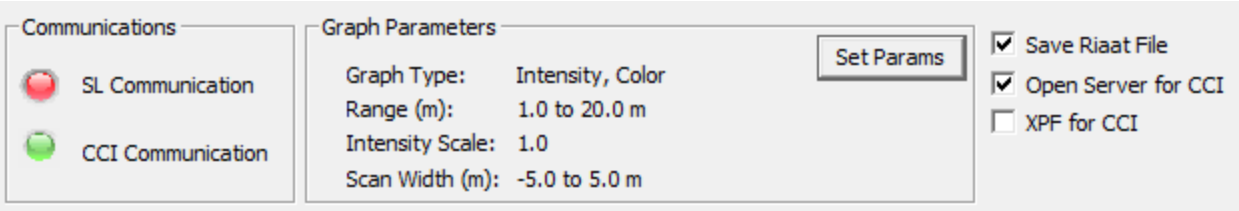
This should get a reply. If using a computer different than self and there is not a reply, then the router path and firewalls need to be checked. If communicating to self and there is no reply, then on windows machines a user might have to run a command prompt as administrator and execute the command:

xxx> netsh http add iplisten 127.0.0.1

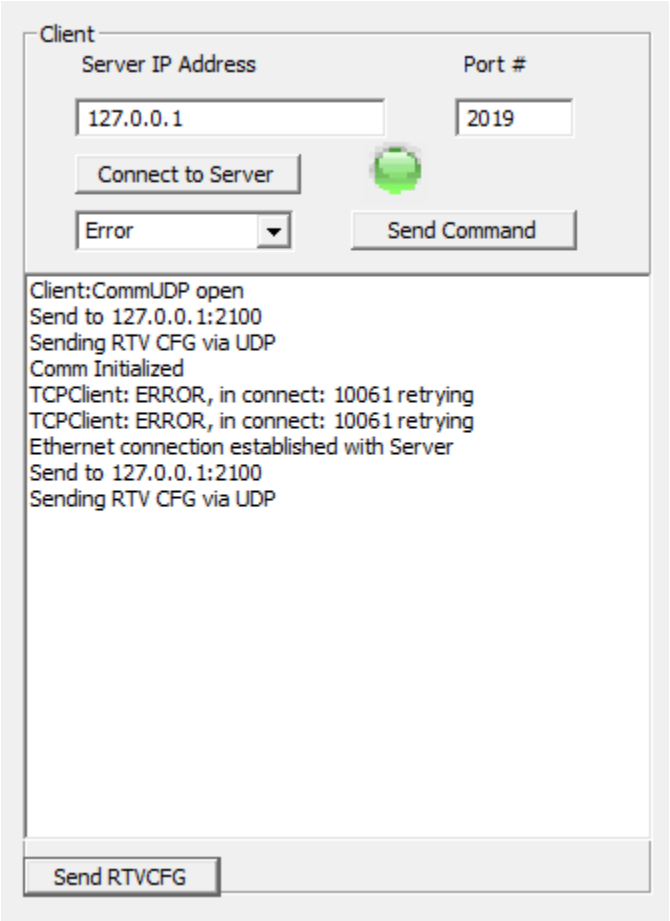
This adds in a listener that will allow connections at this IP address. Test with ping to verify there is a reply now. Firewalls may still interfere with operation and rules may need to be setup to allow specific IP addresses and ports through the firewall and/or for specific programs.

The **Port #** is the TCP IP port number that is used. This is documented in 3DatDepth_CCI_packets_RevC5. Both the sender and receiver must agree on the port for a successful connection. Port 2019 is the default.

Once the IP address and port are agreed to, then hit the **Connect to Server** button. The status light will turn Green if successful. Note in the RTV the box **Open Server for CCI** must be checked! Otherwise RTV will ignore the interface and the light will stay red. What RTV looks like on successful attachment:



Next the **Send RTVCFG** button can be pushed. Should see something like this:



If data is being sent by RTV, the text screen will show packets showing up.

The user can also send commands { **Error, Ack, Report, Reset** } these are options in a pull down menu next to the **Send Command** Button. These act on individual packets or on the interface.

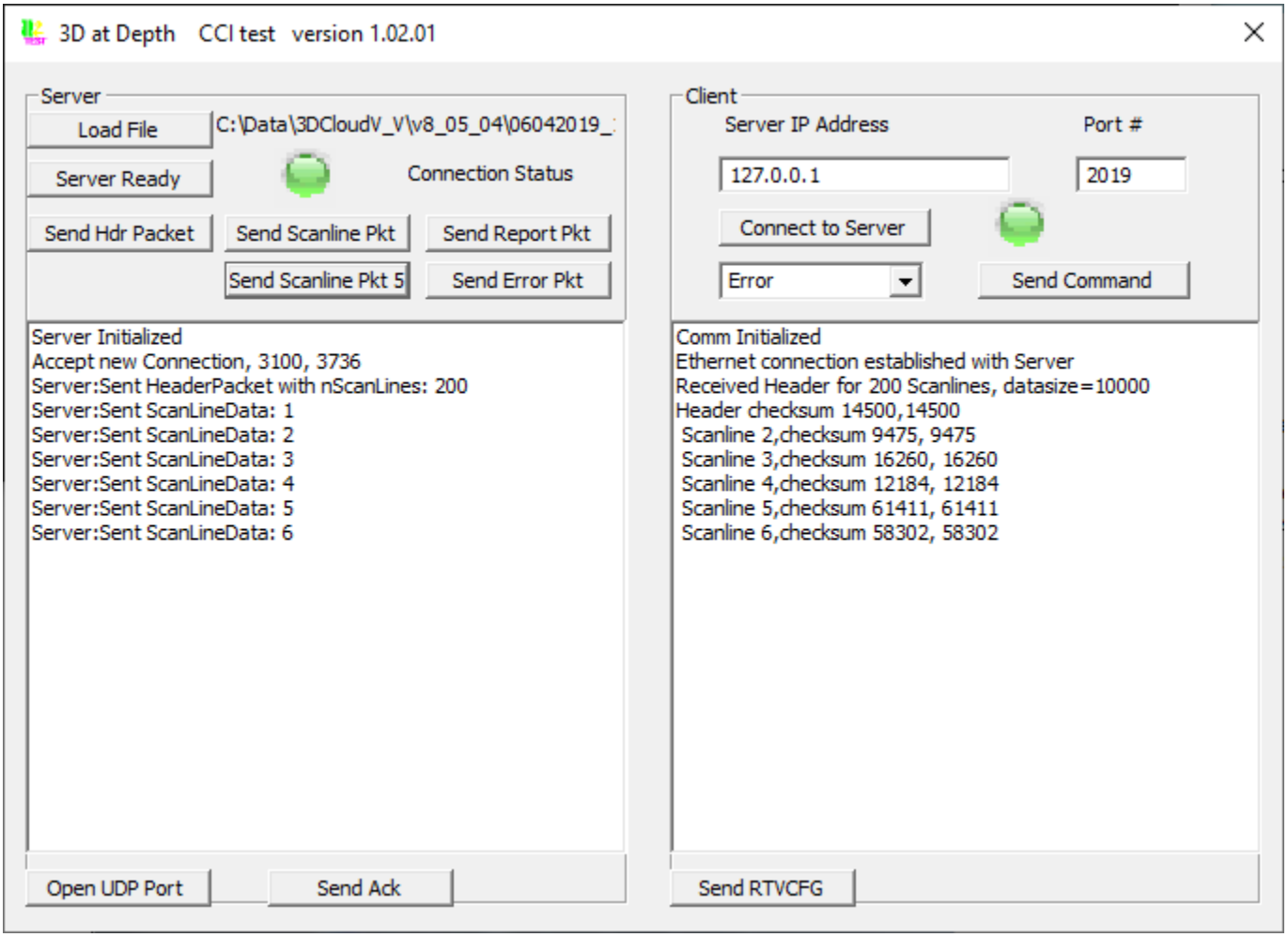
4.3.2 Server Side

The Server side (left side) is the part that will allow a Client to attach and is used to simulate the RTV side that produces the CCI packets. It is possible to have two instances of the CCI test application that attach to each other or to have the server on the left send data to the client on the right.

1. First select a file to replay by hitting the **Load File** button. This will open a browser and allow a user to select an *.RIAAT file to replay.
2. Next hit the **Server Ready** button to startup the server. A message will be reported for Server Initialized and the light will turn yellow.
3. At this point the Client on the right side could be connected (or another user application that has a Client interface can now attach by trying to open up a TCP IP connection on the 2019 port on the computer. This can

be the 127.0.0.1 if the other application is on the same computer or a IP address of a network that the computer is on. Note that the firewall will have to allow connections to/from computer or it may get blocked). The IP address is 127.0.0.1 and port is 2019 for the server on the left. Select **Connect to Server** button and a command may need to be sent to verify it is activated.

4. The **Send Hdr Packet** button should be done first, then **Send Scanline Pkt** button and/or **Send Scanline Pkt 5** button. The difference is that the first only sends one scan line at a time while the second button will send five scan lines when it is pressed. Not every line will be reported in the GUI since it will suppress messages if it falls behind. The attached client reports:



5. Other command options are to send a **Report Pkt** and an **Error Pkt**.

These command options should allow testing of the interface and the ability to receive data. The main difference between the test application and the RTV is that data is often produced very quickly so the packets will likely require buffering and a quick interface to keep up. If the interface is full, then packets will be dropped at the time of sending them. If a packet is corrupted, the client can ask for a packet again, but there is a time limit on how quickly it has to come in since the packet is only buffered for a short time. If the server still has the packet in memory, then the server will send the packet again. Note that this means that the packets will be received out of order since the asked for packet is injected into the data stream.

Once the client is attached to the server, the system should be closed and restarted if a different client wants to connect to make sure all buffers are cleared and reset.