CEP Packet Stream

The CEP packet stream is the lowest level, constantly produced data product from the station. It is a series of UDP packets that provide information on the digitized voltages for a given antenna while a beamformed observaiton is performed. There are normally 4 ports of data produced, though this may be lower if you are not fully utilizing the beamlets available to your station.

The CEP packet stream is highly variable, based on both your observing setup and your RSP configuration (`/var/lofar/etc/RSPDriver.conf`), we will describe this format as generic as ossible and provide default values for an international station.

Generating the CEP Packet Stream

The CEP packet stream should be generated whenever a beamformed observation is started via `beamctl`, though your station ocnfiguration or the use of `rspctl` to create BST data may interrupt or stop the data stream. You can verify the stream is enabled through the `rspctl –datastream` command, and re-enable it by calling `rspctl –datastream=1`.

While enabled, the CEP packet stream will send data to the MAC addresses configured in your `RSPDRiver.conf` every N time samples (by default, 16, 81.92µs) on all ports; though some ports may not contain data if the beamlets are not allocated. The subbands are distributed as described below (values are always inclusive).

Port Offset	4-bit Data	8-bit Data	16- bit Data
0	0:243	0:121	0:60
1	244:487	122:243	61:121
2	488:731	244:365	122:182
3	732:975	366:487	183:243

As the packets are UDP packets, the data may arrive at your machine out of order or not arrive at all. Packet loss to on-site machines is normally relatively low and in-order, so performing operations on the raw recorded data should be fine for more cases.

CEP Packet Data Format

The full CEP packet is described on page 32 of the Station Cookbook, but for now we will focus in on the CEP header, without any of the UDP information, and the raw data

There are 16 bytes of relelvant information at the start of every CEP packet. While it doesn't provide as much metatdata as we wouldl ike, it gives a good starting point for validation the structure of a packet and some of the base observing parameters.

Parameter	Byte(:bit)	Usage
RSP Version	0	Validation (~=3)
Source Info	1-2	Observing Configuration
RSP ID	1:0-4 (5-bit)	Output RSP ID
Unused	1:5 (1-bit)	Validate 0
RSP Error	1:6 (1-bit)	RSP Error, validate 0
Clock Bit	1:7 (1-bit)	1 if 200MHz clock, 0 if 160MHz
Bit mode	2:0-1 (2-bit)	0: 16-bit, 1: 8-bit: 2: 4-bit, 3: ERR
Unused	2:2-7 (6-bit)	Vaidate 0
Config	3	
Station ID	4-5	Station Code (see below)
Number of beamets	6	Beamlets in the current packet
Number of time samples	7	Time samples in the current packet (normally 16)
Timestamp	8-11	Unix timestamp of observation
Sequence	12-15	Leading time sample sequence

The actual packet size can vary based on the observing methodology. You can predict the size of each packet, and the raw data deminsions as a result, from the number of beamlets and number of time samples (bytes 6 and 7).

The reported statino code does not correspond to the public station names, I.e. IE613, SE607, etc. The RSPs report the internal station code, multiplied by 32, with an offset depending on the output port. The full list of station codes can be found here. As an example, if we were to analyse the short value produced from IE613, we would expect to see (214 * 32) + (0,1,2,3) depending on the CEP port analysed.

The data is then in time-majour order, where each time sample contains the (Xreal, Ximag, Yreal, Yimag) samples. Size of each sample depends on your bitmode, varying from half a byte (4-bit) to 2 bytes (16-bit). This repeats for $N_{time\ sampels}$ (default is 16), before moving on to the next beamlet.

Recording / Handling Methodology

This is discussed more in-dept within the REALTA user's guide where we describe the methodology used at IE613. The overall view is that the data should be

- Recorded with some UDP packet capturing software -- wireshark, Olaf Wucknitz's volage recorder, etc.
- Data is read back either blindly or checking for missing packets, out of sequence data, headers analyzed, etc
- Voltages can be used to form Stokes vectors to the output data product required

---

Revision #6
Created 1 July 2020 16:44:43 by David
Updated 6 December 2020 16:14:48 by David