# Data Compression for HERA

Paul La Plante, James Aguirre, Bryna Hazelton, Miguel Morales

## 1. INTRODUCTION

The size of the dataset for HERA will soon become intractable for either storage or computing if some thought is not given, since the rate of data growth exceeds Moore's Law by a large factor, particularly the growth in data volume compared to the decrease in cost of storage media.

## 2. PREVIOUS WORK

Tony Willis and Stephen Wijnholds investigated the effect of using baseline-dependent averaging (BDA) in the context of the SKA (Wijnholds et al., 2018). They did the work separately and it was simulations for SKA configurations (they were on a committee that was looking on it and they ended up working independently). Their analysis showed that one can achieve very high data compression rates (upwards of 80%) when using BDA while minimizing the induced decorrelation when imaging. However, their analysis was in the context of a tracking antenna array, and so is not wholly applicable to HERA. Nevertheless, BDA is a promising method for decreasing data rates.

## 3. CORRELATOR BUILDOUT

The correlator will have several different incarnations over the coming year. We recap the main features, and the resulting data rates assuming no compression.

| **Observing Season** | $N_{\text{channels}}$ | **Integration time** | **Raw Data Rate** | **Nightly Data Volume** |
|---|---|---|---|---|
| **H2C.A** | 1536 | 10s | 0.045 GB/s | 1.94 TB |
| **H2C.B** | 3072 | 10s | 0.346 GB/s | 14.9 TB |
| **H3C** | 6144 | 2s | 12.1 GB/s | 523 TB |

Note that in the above table, additional antennas are also being added during different campaigns, so the data rates and volumes are not simply a factor of 2 larger when the number of channels is doubled. Also, for H2C.A and H2C.B, the data rates corresponding to the maximum number of antennas are quoted (December 2018 and March 2019, respectively). Clearly, heavy averaging must be done by the start of H3C. Even by H2C.B, significant averaging will be necessary, and raw data will likely not be saved in perpetuity.

## 4. CONSTRAINTS

In addition to the sheer volume of data the correlator will be generating, there are further constraints being placed on the storage and transfer of data.

### 4.1 On-site storage

Currently, there is ∼550 TB of storage on-site in the Karoo. To satisfy the plan laid out below, further storage will be required. At the NRAO, the cost of storage works out to roughly $75,000/PB. We assume a comparable cost for on-site storage as well, but with shipping costs are likely to be closer to $80,000/PB.

### 4.2 Network Bandwidth

We are actively working with SANReN and the SKA in South Africa to secure additional bandwidth. However, assuming that no improvements are made, the current throughput is ∼200 Mbps. Restricting ourselves to transferring data only during off-peak hours, this translates to roughly 1 TB of data volume that can be moved per day.

For the coming season, we are investigating leveraging a data transfer node (DTN) operated by SKA to speed up data transfers to NRAO. If we are able to sustain 380 Mbps (with 100% uptime, or 760 Mbps for 12 hours), we can transfer 1.5 PB of data per season to NRAO. Real-world testing is still required for determining what throughput is possible.

### 4.3 Archival Storage

The NRAO currently has ∼384 TB of storage. Under the current MSIP, there is funding earmarked for purchasing another ∼500 TB of storage. Under the newly funded MSIP, the plan is to purchase a total of 2 PB of storage at NRAO. Unfortunately, due to space and power constraints at the NRAO processing facility, there is no opportunity to grow beyond 2 PB.

Simultaneously, there is the possibility of acquiring significant storage space at Cambridge University. Preliminary discussions suggest up to an additional 1 PB may be possible, though it remains to be seen how much can be promised to HERA.

## 5. PLANNED COMPRESSION TECHNIQUES

For context, the goal is to provide compression and averaging techniques that will reduce the nightly H3C volume of data to 50 TB (down from ∼500 TB). This reduction will make the volume of data manageable (though still quite large for an entire season).

### 5.1 Bitshuffle

Bitshuffle is a filter that can be applied to data saved in the HDF5 format. It rearranges data to allow for highly efficient compression for certain data types and data applications. The bitshuffle algorithm was originally developed for and applied to data in the CHIME pathfinder telescope (Masui et al., 2015). There, bitshuffle achieved a compression ratio of 4:1 (meaning the compressed data was 4 times smaller than the original uncompressed data). More conservatively, we assume a **compression ratio of 2:1** (factor of 2 savings). Bitshuffle is a totally lossless compression method, and so can be safely applied independent of any other averaging schemes.

One potential issue with bitshuffle is that it has not yet been implemented on the HERA data from the new correlator, and so there is some uncertainty as to the exact amount of savings that will be had. One limitation of bitshuffle is that it works best for integer data types. Though the plan is for the new correlator to write out integers, some study needs to be done to ensure this will not bias our answers in the presence of fringe stopping. If the data from the correlator is saved as floating point numbers instead, the savings from bitshuffle are more modest (David MacMahon reports ∼15% savings at a different radio telescope project).

### 5.2 Baseline Dependent Averaging

As discussed above, BDA has been explored in the context of the SKA. There, as here, the averaging was proposed to be done in time. (Averaging in frequency is also possible, but scary, because we do not want to impart any artifacts along the frequency axis.) The general scheme is to average adjacent integrations in time until a critical threshold is met (determined by the amount of decorrelation induced when the resulting visibilities are used to generate images). In effect, different baseline lengths (and orientations) have different integration times, and longer effective integration times decreases the total data volume. For a fixed amount of decorrelation, shorter baselines can be averaged for more time before reaching the decorrelation threshold. Due to the nature of HERA's layout, and the fact that the majority of baselines are relatively short, there is a significant potential savings from BDA. The tradeoff is that *all* baselines in a given dataset will have that level of decorrelation, not just the longest baselines.

Preliminary calculations show that for the correlator specifications defined for H3C, a **compression ratio of roughly 12:1** (92% savings) is possible. The threshold chosen for the calculation is a maximum decorrelation of 5% for a source that is 20° off of zenith. There is also a maximum averaging time of 30s, to mitigate the effect of sources moving inside of the beam (because we are not a tracking array, unlike the SKA). Interestingly, allowing for more decorrelation does not lead to significantly more savings, and so this level is a "sweet spot" for trading off decorrelation and data compression. Note that testing of the BDA scheme with imaging tests, as well as the software necessary to support this scheme, must be developed.

### 5.3 Resulting Nightly Data Volume

The goal is to reduce the nightly data volume from ∼500 TB to 50 TB. The bitshuffle is projected to provide a factor of 2 in savings, and BDA is projected to provide another factor of 10. The total savings is thus a factor of 20. This results in a nightly (raw) data volume of 25 TB. Accounting for additional ancillary files produced by RTP (e.g., calibration solutions, RFI flags, etc.), the total nightly data volume should remain under 50 TB.

It should be noted that 50 TB per night of observation is still a tremendous amount of data. A full season's worth of this data (6 months) is nearly 9 PB, which we simply *cannot* save in its entirety. As such, we must further average the data into smaller, more manageable datasets. The total amount of data we could realistically save is 2 PB, but we will limit ourselves to 1.5 PB as a more modest goal.

Below, we propose several additional averaging methods that can be applied. Because these are not lossless (like bitshuffle) or nearly lossless (like BDA), there are various assumptions and downsides that are present. They are also better suited to different analysis techniques and use cases, and provide a level of protection against unforeseen issues that may be present in the other methods.

## 6. PROPOSED ADDITIONAL AVERAGING METHODS

### 6.1 LST binning on-the-fly

One way to decrease the data rate is to LST bin the data as it is taken and calibrated. Essentially, there is a single LST grid laid out on disk, and as data has been processed, it is added to the corresponding bin. This leverages the stability of the instrument and the sky night-to-night to combine data that should nominally be the same (or close).

**Proposed Data Volume**

- 50 TB (base nightly amount) × 1.5 (for 18 hours of LST coverage instead of 12) × number of LST-binned datasets. For example, we can accumulate the LST binning in 10-day chunks, so that if something happens to contaminate one dataset (unflagged RFI, poor calibration, etc.), not all of the data is affected.
- Assuming 500 TB of available space, this corresponds to 6-7 such 10-day datasets.

**Assumptions**

- Relative calibration error from night-to-night is well-handled.
- Precession of the equinoxes over the LST-averaging window is negligible.

**Potential Downsides**

- Requires an initial calibration before data can be LST binned.
- Might not be able to perform antenna-based calibration in post-processing once averaging has been done (that is, redundant baselines may no longer be "redundant enough" to determine an overall single antenna-based gain).
- Requires a "Bayesian update" step that decides whether to add a given night's integration into the running total. The data should only be added if the data in question is "close enough" to the existing data to avoid introducing errors.

### 6.2 Redundant Baseline Averaging

A more agressive way to decrease the data volume is to average nominally redundant baselines together. For instance, all 14.6 m E-W baselines would be averaged, and a single output baseline would be produced.

**Proposed Data Volume**

- The ratio of total baselines to redundant ones is ∼10:1, so the final product would be ∼5 TB/night.
- Assuming 500 TB of available space, roughly 100 days (almost a whole season) can be stored.
- By further downselecting the data (e.g., keeping only short baselines and clean frequency windows, and forming pseudo-I Stokes visibilities), the data volume may be small enough to send this product back nightly for running power spectrum analysis.

**Assumptions**

- Nominally redundant baselines are actually redundant.

**Potential Downsides**

- Can image the resulting dataset, but can no longer perform antenna-based calibration.

- In particular, the beams of antennas would need to be similar enough (conservatively a part in $10^5$) that we are not introducing artifacts and can still perform precision calibration. We know from current data that redundant baselines are not redundant at the few percent (or more) level.
- Average beam is different between different baselines, and so the resulting visibility is an average weighted by the number of times an antenna appears in a given baseline group. This could be Very Bad if there is a misbehaving antenna that deviates significantly enough from the mean to move the mean value.

### 6.3 Imaging-Based Analysis

Using the data to generate images of the sky (more correctly, image-cubes, where frequency maps into line-of-sight distance/redshift) is another method for decreasing the total data volume. However, at intermediate stages, the data requires significant reduction, due both to general storage restrictions and to facilitate real-time analysis. Thus, there are specific ways to reduce the amount of data stored:

- Narrow the LST coverage of imaged datasets. By restricting data to, say, 3 hours of LST coverage, significant savings compared to the raw data generated can be had.
- Preserving only a subset of the frequency range at full time/frequency resolution for imaging purposes. For instance, we can entirely cut below 108 MHz if we do not think we can reliably image the EoX. We could also consider saving a subset of frequencies (picket fence) outside a prime observing window to provide lower sensitivity leverage for calibration and foreground subtraction.
- Only retain linear polarizations `xx` and `yy` for generating Stokes I images for most times, and only keep the cross-pols for an estimate of thermal noise (which requires a much smaller amount of data to be saved).

## 7. PROPOSED PLAN

The proposed plan for the incoming deluge consists of several parts. We outline here the main points.

### 7.1 Hardware Changes

Before H3C, there are several hardware changes that need to be made.

- Purchase and additional 1 PB of storage on-site ($\sim$\$80,000). This will bring the total amount of on-site storage to $\sim$1.5 PB.
- Install more networking switches in the KDRA, to facilitate faster communication between the data storage nodes and RTP processing nodes.
- Create a data transfer node (DTN) to provide for more reliable and faster network transfers of data to the NRAO.

### 7.2 Software Changes

In addition to hardware changes, there are significant software endeavors that must be undertaken.

- Implement the software for performing BDA. This should be in place well before BDA is planned to be implemented. It must be tested not only for profiling the amount of compression that is possible, but more importantly to understand the effect that BDA has on down-stream analysis, in particular imaging.
- Build support for interpreting BDA-compressed datasets using high-level tools (e.g., `pyuvdata`)
- Develop software for constructing LST-averaged datasets and redundancy averaged datasets. The latter will also likely require high-level software support for interpretation.
- In H3C, RTP will likely be running multiple pipelines simultaneously for different science analysis. For instance, there may be a delay-spectrum pipeline and an imaging one. This model will require coordination between the science teams about what data will be taken, how to process it, and what output products will be generated. Much more detailed discussion will take place before the H3C observation season.

### 7.3 Storage Allocation

As mentioned above, there will be a total of 1.5 PB of storage space in the Karoo. We provide here a summary of how that space will be partitioned.

- 400 TB: LST-averaged datasets. As outlined above, this amount allows for 4-5 datasets. If the LST-averaging window is 10 days, this is roughly 1.5 months' worth of data.
- 350 TB: Redundancy-averaged datasets. We are projecting the final redundancy-averaged datasets to be roughly 5 TB. This allows for ∼75 days to be saved (2.5 months of data).
- 750 TB: Datasets which have been earmarked for imaging, as well as finished products from imaging analysis. Research will be required to determine what imaging products will be produced, as well as the data input required for this analysis.

There is some freedom to move data allocations around between the uses outlined above, but note that the total volume must stay below 1.5 PB.

### 7.4 Nightly Processing Plan

Given all of the above, we outline here what A Day in the Life of RTP looks like.

1. Record data from correlator at full time and frequency cadence, yielding ∼500 TB of raw data (final volume may be a factor of ∼2 smaller on-disk by using bitshuffle).
2. Perform QM analysis/RFI flagging on full-resolution data.
3. Apply BDA scheme to yield 50 TB of raw data.
4. Perform antenna-based calibration, and accumulate in LST-binned files (with each LST-binned dataset being ∼75 TB)
5. Generate redundancy-averaged dataset (∼5 TB/night)
6. Perform imaging-based analysis on data taken. This will likely be on a reduced subset of the data, which has been downselected in frequency or LST range in some fashion.

## References

Masui, K., Amiri, M., Connor, L., et al. 2015, Astronomy and Computing, 12, 181

Wijnholds, S. J., Willis, A. G., & Salvini, S. 2018, MNRAS, 476, 2029