

VLBA Sensitivity Upgrade Memo 54
X-cube to Data Buffer Server Transition

Walter Brinken

25 Oct 2022

Abstract This memo aims to document the initial state, using the original X-cube switches to explore and transfer RDBE baseband data, and its transition to the final state, using the new Data Buffer Server in the VLBA New Digital Architecture context.

1 Introduction

The VLBA New Digital Architecture (VNDA) project is underway and will replace the ROACH Digital Back Ends (RDBEs) with a new system. The X-cube computer plays an integral role in the RDBE system, performing the following functions:

- Merging the output data from two RDBEs for input into the Mark5C, and later Mark6, recorder
- Extracting pulse calibration tones for real-time diagnostics¹
- Capturing small amounts of data for near-real-time use
- Forwarding data in real time off-site over high-speed networks

The X-cubes are old by computing standards, being installed as part of the RDBE system in 2011. At the time of their installation, when they fed the Mark5C recorders, they relied on a particular network interface card (NIC) and modified kernel driver to perform “packet pacing”, preventing the shallow input buffer of the Mark5C from being overwhelmed. The Mark6 uses a standard Linux network stack and does not share the same shallow buffer issue, so does not rely on special hardware and drivers.

1.1 Current status

Hardware to replace the X-cube systems was procured in 2021. As of June 2022 the equipment, a new operating system (RHEL8), and updated software suite is in testing on the VLBA test rack. The first Data Buffer Server (DBS) was successfully installed at Pie Town on Oct 13, 2022. The goal is for the remaining X-cube systems to be deployed before Jan 1, 2023.

1.2 Document layout

The document is laid out as follows. Sections 2 through 6 each describe a phase of the transition plan. Each section introduces the changes relative to the previous. New hardware and software components are described as they are introduced.

Section 7 provides some details of some of the software interfaces.

Note that not every bit of hardware or software involved in RDBE/VNDA operations at the site is described in this document. The central focus of this document is the DBS and the elements that interface to it.

¹The DiFX correlator will extract the tones again at correlation time and has the capability of capturing an arbitrary number of tones.

2 Phase 1: Initial state

This section describes the starting point of the transition. With the exception of the Mark6 recorder replacing the Mark5C recorder and the introduction of the `sendit` program, this essentially represents the end state of the RDBE development that concluded approximately a decade ago.

2.1 Hardware configuration

The initial state consists of the following equipment:

Name	hostname	primary IP address
Control computer	<i>stn-cc</i>	10.81. <i>subnet</i> .11
RDBE 1	<i>stn-dbe-1</i>	10.81. <i>subnet</i> .12
RDBE 2	<i>stn-dbe-2</i>	10.81. <i>subnet</i> .13
X-cube	<i>stn-xcube-1</i>	10.81. <i>subnet</i> .20
Mark6 recorder	<i>stn-mark6-1</i>	10.81. <i>subnet</i> .27
R-rack PDU	<i>stn-pdu</i>	10.81. <i>subnet</i> .252

Here, and in following sections, the three-digit *subnet* is determined by the station code (*stn*) according to the following: *sc*=101, *hn*=102, *nl*=103, *fd*=104, *la*=105, *pt*=106, *kp*=107, *br*=108, *ov*=109, *mk*=110. The hostname and primary IP addresses provided above correspond to the main login and M&C route. The RDBEs and X-cube systems have other special-purpose networks as well. Equipment on the VLBA test rack operates with *stn*=vlb1 and with IP addresses of the form 10.64.94.*xx*.

A diagram showing how all of the equipment is connected is shown in Fig. 1. A summary of some of the relevant hardware follows.

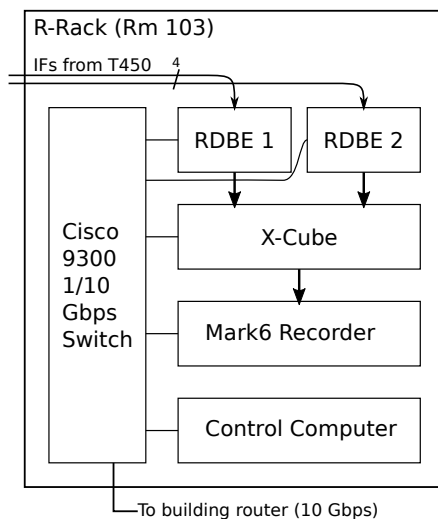


Figure 1: *Configuration of equipment in the initial configuration.* Thick arrows show path of full-data-rate science data. Thin lines show monitor and control network. Analog data from the T450 “matrix switch” enters on the two thin arrows near the top; each of these arrows represents two if cables. When performing real-time correlation or data transfer, limited-data-rate science data streams from the X-cube, over the monitor and control network, and then off-site via the building router.

2.1.1 RDBE

Each VLBA station has two ROACH Digital Back End modules. Each of these contains a ROACH (version 1.2 or 1.3) FPGA board, a sampler module, a clock synthesizer, and an attenuator module. Collectively the two RDBEs can sample four analog IF signals in the 512-1024 MHz frequency range. As deployed, the installation can process up to 1024 MHz of bandwidth (512 MHz per polarization in dual-polarization mode) with a maximum output data rate of 4096 Mbps. The RDBEs always produce data quantized to two bits per sample. The output data is real-valued voltage samples formatted either as Mark5B frames (for PFB personality) or 5032 byte VDIF frames (for DDC personality).

2.1.2 X-cube switch

The X-cube systems are essentially well-provisioned 2010-era Linux/x86 servers. These devices were built by XCube Communications, a small R&D company that focused on high speed storage. The case is custom designed, but all internals are standard PC/Server components. There is some variation amongst the deployed units, but generally they have the following typical attributes:

- Chassis: custom 5U rack-mount, painted blue
- Intel Xeon E5620 processor: 2.4 GHz, 8 cores, SSE4.2 SIMD instruction set, 12MB cache memory, 32nm process, 80W TDP, introduced in Q1 2010.
- 12 GB RAM
- 250 GB boot/OS disk
- 3 TB data disk
- 1 Gbps Ethernet (eth0: M&C, login)
- Four 10 Gbps Ethernet (eth2, eth3: inputs from RDBE, eth4 output to Mark6)

2.2 Software configuration

A diagram showing the logical connectivity of the software components is shown in Fig. 2. Especially relevant software elements are described in the following sections.

2.2.1 Executor

An instance of the Executor runs on the Control Computer at each VLBA site. It reads a `.py` file describing the sequence of actions needed to perform an observation and sends commands to the VLBA equipment.

2.2.2 sendit

The `sendit` program is used to route and in some cases modify baseband data. It exposes a VSI-S interface on a TCP port and an HTTP interface for determining status. Because up to three copies of `sendit` are anticipated to run on one machine during the transition period, different ports are being assigned to keep them separate:

Instance	VSI-S	HTTP
Default	2987	2998
RDBE to Mark 6	2986	2997
VNDA to Mark 6	2985	2996

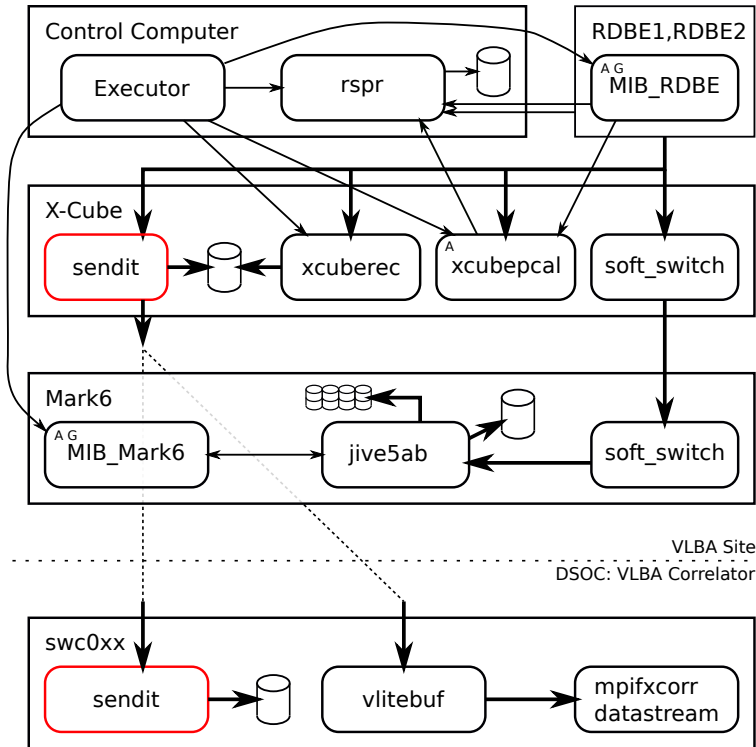


Figure 2: *Connectivity between software elements in the initial configuration.* The software components (round-cornered boxes) in red are in current development. All others are well established. Big arrows indicate flow of science (e.g., VDIF) data. Small arrows indicate flow of monitor and control data and calibration data (switched power and pulse cal). Software components with a small ‘A’ in the top left emit alerts. Software components with a small ‘G’ in the top left are exposed by the operator GUI. Cylinders represent persistent storage (e.g., hard drives) that temporarily store calibration or science data. The computer “swc0xx” represents one of the processing nodes on the VLBA DiFX correlator. During real-time operation each site will send data to one of the computers swc001 . . . swc010. When performing real-time correlation, that machine will host the “datastream” DiFX process as well as a “core” process which performs the cross correlations for all baselines. When performing real-time data transfer, the data is simply stored on a local filesystem.

The Default instance will be responsible for all configurable use cases (e.g., routing data off site or storing data locally), whereas the other two instances will be started in a fixed configuration.

Incoming baseband data can be collected through raw reads of an Ethernet device (as required for RDBEs), from a uni-cast UDP port (the usual case for receiving real-time data), or from a multi-cast UDP source (as will be the case for VNDA equipment).

In addition to routing data, `sendit` can convert Mark5B formatted data to VDIF and can select which channels to propagate.

`sendit` does not have a large RAM buffer so it depends on fast local storage when storing incoming data locally. This drives the decision to use SSDs rather than spinning disk.

2.2.3 `xcuberec`

The `xcuberec` program saves a portion of streaming baseband data to local disk based on the project code. This program is used to support the USNO daily “intensive” observations. The incoming data are stored in

a large (multi-GB) RAM buffer before being written to disk, allowing full rate (4 Gbps) data to be stored on a single spinning disk. The size of the RAM buffer determines the maximum length of full rate data that can be stored using this mechanism. The ratio of disk write speed to incoming data rate determines the maximum recording duty factor.

`xcuberec` receives its incoming baseband data through raw reads of Ethernet devices.

2.2.4 `rspr`

`rspr` captures binary data packets emitted by the RDBE which contain switched power and total power information. It formats this data, combining it with the known state of the RDBE, and writes switched power record files (files with extension `.switched.power.xz`). A new file is created at each UTC midnight, and whenever a project either starts or stops. The Remote Monitor Data Collector (RMDC) program transfers completed files to a computer called “elbow” at the DSOC and places them in the directory `/home/vlba/metadata/switched.power`.

`rspr` performs three other critical functions:

1. It captures binary pulse cal data created by `xcubepcal`, formats them, and writes them to files with extension `.xcube.pcal.xz`. Like for switched power data files, a new file is created at UTC midnight and project start/end. These files are captured by RMDC and placed on “elbow” in `/home/vlba/metadata/xcube.pcal`.
2. It analyses in real time total/switched power being produced during pointing observations. Determined pointing offsets are broadcast to the Executor, which may be applied as part of referenced pointing.
3. It serves switched power data, pulse cal data, and state count data on request via a UDP port. This is mainly used by the `vlba_calserver` program

2.2.5 `xcubepcal`

`xcubepcal` processes the Mark5B- or VDIF-formatted baseband data broadcast by the RDBEs and determine the phase and amplitude of the pulse calibration tones that are injected into the VLBA signal path at the receiver. The resultant “extracted” tones are delivered to `rspr` for formatting and staging for transfer to their archive location on the computer called “elbow” in directory `/home/vlba/metadata/xcube.pcal`. These calibration values aid in delay calibration. `xcubepcal` receives its incoming baseband data through raw reads of Ethernet devices.

2.2.6 `soft_switch`

`soft_switch` was originally written to satisfy the very picky timing needs of the Mark5C data recorder. It takes the output of two RDBE data streams and merges them into a single stream, while “pacing” the output so the Mark5C’s very small input buffer would not overflow. The Mark 6 recorders no longer have this input buffer limitation, but this software is still used instead of a regular Ethernet switch. Two instances of this program are used in the current (initial) configuration. The first resides on the X-cube switch. It accepts baseband data through raw Ethernet device reads from the two RDBEs and outputs both into a single UDP stream. The second resides on the Mark6 recorder. It accepts the UDP baseband stream and is used to ensure Mark5B formatted data is complete. Mark5B data frames are 10016 bytes long and are split into two 5008 frames to fit into jumbo (9000 byte) Ethernet packets. Loss of one of the two packets can lead to a corrupted datastream. The output of this second instance is sent over a local socket to the `jive5ab` recorder software.

Neither instance of `soft_switch` requires configuration once started.

2.2.7 vlitebuf

`vlitebuf` is a program developed for the VLA Low-band Ionospheric and Transient Experiment (VLITE) to feed a real-time DiFX correlator with a stream of VDIF-formatted data. This program listens to incoming data on a raw Ethernet device (for VLITE) or on a UDP port (as used by the VLBA) and stores several GB (a few seconds) of data in a RAM buffer. This buffer is exposed through the Linux FUSE (Filesystem in USER-space) mechanism. It allows an unmodified version of the DiFX correlator to reliably correlate a real-time data stream with a few seconds of buffer to accommodate start-up latency and differential delays.

In its use by the VLBA, a controlling program, `vlitebufmgr`, exposes a VSI-S interface that allows some configuration. This program starts and stops `vlitebuf` based on commands from a managing process.

2.2.8 jive5ab

`jive5ab` is the primary software that runs on the Mark6 recorder. It receives data on a UDP port (which on the VLBA is fed by `soft.switch`) and streams the data to a hard disk module.

3 Phase 2: DBS deployment

In phase 2, the decade-old X-cube switch hardware will be replaced with modern servers.

3.1 Hardware configuration

Name	hostname	primary IP address
Control computer	<i>stn-cc</i>	10.81. <i>subnet.11</i>
RDBE 1	<i>stn-dbe-1</i>	10.81. <i>subnet.12</i>
RDBE 2	<i>stn-dbe-2</i>	10.81. <i>subnet.13</i>
Data Buffer Server	<i>stn-dbs-1</i>	10.81. <i>subnet.20</i>
Mark6 recorder	<i>stn-mark6-1</i>	10.81. <i>subnet.27</i>

The DBS will reuse the IP address from the X-cube. It will not be possible for both an X-cube and a DBS to be operational at a VLBA site at the same time.

3.1.1 Data Buffer Server

The Data Buffer Server (DBS) is essentially a modern update to the X-cube. It is a Commercial Off-The-Shelf (COTS) commodity purchased from Dell.

- Chassis: Dell PowerEdge R340 1U rack-mount
- Intel Xeon E2286G processor: 4.0 GHz, 6 cores, SSE4.2, AVX2 instruction sets, 12MB cache memory, 14nm process, 95W TDP, introduced Q2 2019.
- 64 MB RAM
- To be configured for remote boot, NFS mounted OS partition
- Two 4 TB SSDs in software RAID0 for data
- Two 1 Gbps Ethernet (eth0: M&C, login)
- Two 10 Gbps Ethernet (eth2, eth3: inputs from RDBE in initial configuration)
- One 10 Gbps Ethernet (enp3s0f0np0, output to Mark6 in initial configuration, bidirection connection to VNDA switch in final configuration)

- One additional 10 Gbps Ethernet that may substitute for the 1 Gbps M&C link to allow wide-band off-site data transfer.

Note that the Ethernet device names (`eth0, ...`) are subject to change. 40 Gbps Ethernet devices will likely replace two of the 10 Gbps devices in the late stages of deployment to allow greater recording data rates. These device names and other configuration will be set at boot time by a script².

3.2 Software configuration

In the first deployment of the DBS, its functionality (e.g., software and interfaces) will be identical to that of the initial state with the exception that the `soft_switch` program instances will be replaced with a single instance of `sendit` running on the DBS. `sendit` will convert Mark5B formatted data into VDIF packets, eliminating the need for the `soft_switch` that runs on the Mark6 unit. Downstream software (e.g., software that prepares correlator jobs) will need to be aware that the recorded data format is VDIF, even when initially configured as Mark5B.³ This instance of `sendit` will run with a fixed configuration, pushing data to the receive port (29000) on the Mark6 unit. All baseband data will be VDIF packets with 5000 payload bytes. The DBS will adopt the IP address used by the X-cube.

A second instance of `sendit` will be running on the DBS. This version will be configured by `rtvlba` (but eventually `dbctrl1`) to control the flow of real-time data offsite. This is unchanged from the initial configuration.

Some software optimization work may take place during this phase, such as updating the pulse cal extraction to make use of the more performant AVX2 instruction set. The `openrc` initial system currently in use on the X-cube systems will be replaced by `systemd` on the DBS systems. Minor changes will be needed to `xcuberec`, `xcubecal` and `sendit` to support this.

For completeness, the phase 2 software configuration is diagrammed in Fig. 3.

3.3 Roll-out

The new DBS hardware and its “diskless” (in the sense of Boot/OS) configuration will be tested on the VLBA test rack. Once the new configuration is stable, it will be rolled out to one site on a double maintenance day and will be tested in its primary use case (regular observing). After approximately one month of successful tests, the other sites will be upgraded.

If possible, name resolution will, at least initially, resolve the X-cube name to point to this address as well, but once this phase is complete, the X-cube name resolution will no longer be supported.

Roll-out will happen one station at a time, at least initially. After the first DBS is deployed at Pie Town, the old X-cube unit will be returned to Socorro. The CX4 network cards will be removed and used to populate additional DBS units.

3.4 Impact to other systems

In this phase 2 configuration, all data being produced by the PFB personality in Mark5B format will be converted, on-the-fly, to VDIF format prior to being recorded on the Mark6 system. MarkB data saved locally using `xcuberec` will not be converted. This converted data can be handled by downstream software, but a minor change to `oms2v2d` will be needed to generate the appropriate configuration statements in the `.v2d` file. These changes can be made by hand (by data analysts) if needed for testing.

3.5 Acceptance testing

Final development of software and testing will take place on the VLBA test rack in the DSOC. Two RDBEs and a Mark6 unit will be connected to the DBS. A local copy of the Executor, running on one of the

²<https://gitlab.nrao.edu/nms-vlba/dbs-net-config>

³Initially this can be handled by just changing the correlator job files (`.v2d` files). Later `sched` will be modified to request VDIF data and `vex2script` will appropriately respond to that.

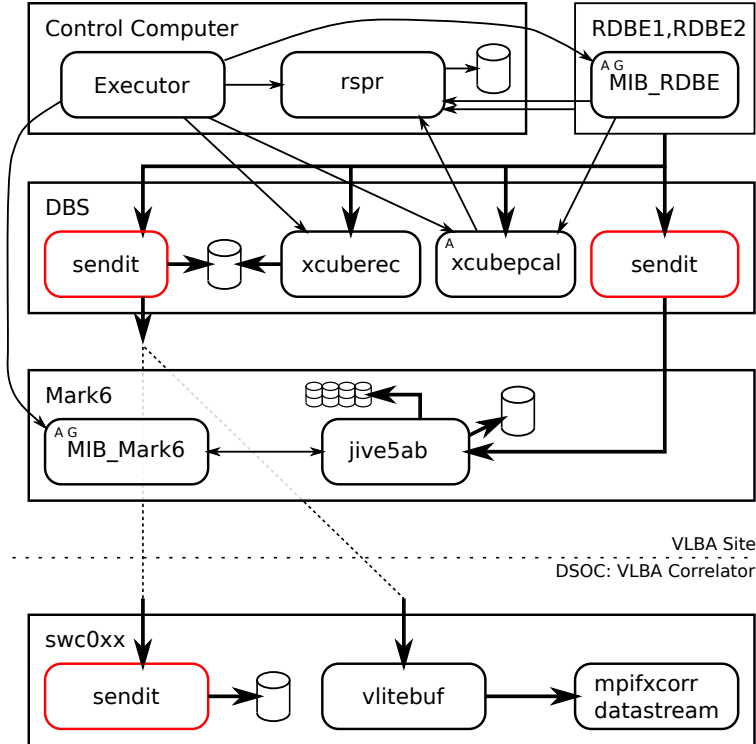


Figure 3: *Connectivity between software elements in the phase 2 configuration.* The symbology follows that used in Fig. 2. The primary change is replacing `soft_switch` with `sendit`

control computers (likely `vlb1-cc`), will execute three test scripts: `short.pfb.VL.py`, `short.ddc.VL.py`, and `short.ddc.2DBE.py`. The recordings made on the Mark6 will be investigated for completeness and correctness. Tests of `xcuberec` and `xcubepcal` to ensure they perform as expected should be made as well.

3.6 Enabled capabilities

The conversion from Mark5B to VDIF before recording will enable two previously impossible, but likely useful, modes to be developed:

- A dual-PFB mode producing 32 baseband channels, each 32 MHz wide. This would enable a 4 Gbps mode which can be used in cases where careful amplitude calibration is required.
- A mixed PFB and DDC mode, allowing 20 baseband channels, each 32 MHz wide. Four of the output channels would have flexible tuning. This could be of value to geodetic observing.

Development of these new modes is not expected to be a significant amount of work, with `Sched` and `vex2script` requiring the largest effort.

4 Phase 3: Continued software modernization

In phase 3, the `sendit` and `dbctrl` programs will replace the functionality of `xcuberec`.

4.1 Hardware configuration

No hardware changes will occur in phase 3.

4.2 Software configuration

The functionality of `xcuberec` will be absorbed into `sendit` and `dbctrl`. Two instances of `sendit` will be running on the DBS. The first one will prepare data for recording by `jive5ab`. The second copy of `sendit` will be used for recording data (at full rate) to local storage on the DBS for later transmission off-site, or for sending data (at data rates possibly limited by the network links) off-site in real time. All commands to X-cube MIB are moot at this point and can be removed from `Executor`, but the `.py` scripts may continue to produce the X-cube configuration, so these should still be accepted in the `.py` files.

The `Executor` will need to be modified to send “recorder-like” commands to `dbctrl` based on appropriate intents that are included in the `vex` file. A minor change to `vex2script` may be needed to pass these intents.

The software connectivity after completion of phase 3 is shown in Fig. 4.

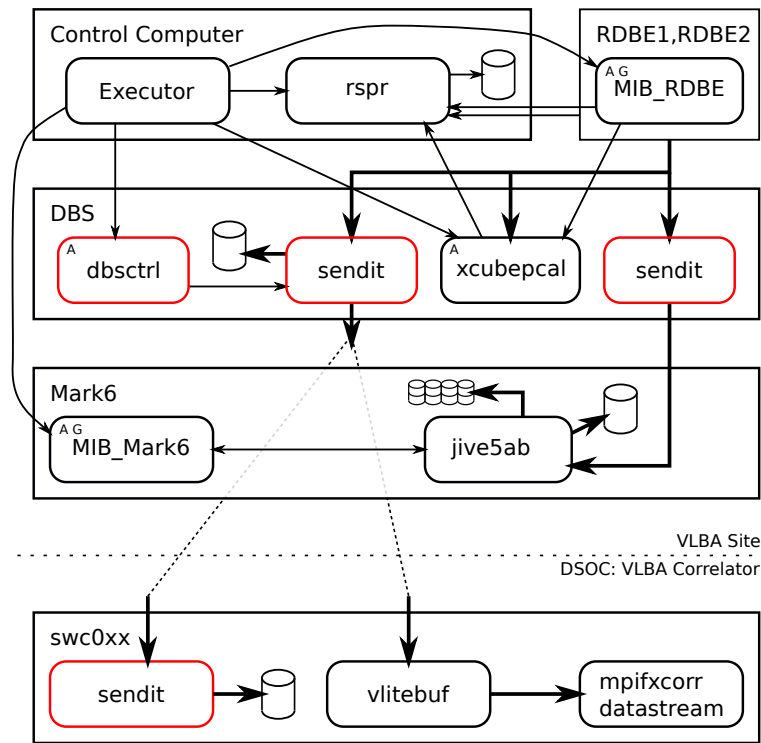


Figure 4: *Connectivity between software elements in the phase 3 configuration.* The symbology follows that used in Fig. 2. The primary change is integrating `dbctrl` into the system and using that to control the instance of `sendit` that performs real-time data transmission. That instance of `sendit` will also subsume the role that `xcuberec` played.

4.2.1 dbctrl

`dbctrl` will control software in the data transmission path based on commands provided by the `Executor`. The software in question includes the `sendit` instance running on the DBS, a `sendit` instance running on a VLBA correlator server, and a `vlitebuf` instance running on a VLBA correlator server. `dbctrl` will

emulate a MIB interface. A separate document is in development that describes in detail what `dbstr1` does and how it interfaces to `sendit` and `executor`.

4.3 Roll-out

The software changes impact the primary data path. These changes have more opportunity to disrupt science observing than the phase 2 changes. The updated software will be tested on the test rack. Once the software changes are validated, one or two sites will temporarily switch to the new software configuration for test observing. When testing has shown satisfactory results, one site will transition to full-time phase 3 configuration, probably for about one month. Data from this site will be examined for data completeness. Once the system is fully validated at one site, the other sites will see the same roll-out.

4.4 Impact to other systems

All observing programs that require use of `xcuberec` will need to have their configuration transferred over to `dbstr1` which will drive the “data transfer” instance of `sendit` to perform the equivalent functionality. The resultant filenames will differ, causing some software updates to users of such data.

4.5 Acceptance testing

Testing of `sendit`’s capability in replacing `xcuberec` will require execution of test projects that closely resemble those projects that require use of `xcuberec`. Initial testing can occur in a simulated environment on a workstation. Final testing will be performed on actual hardware at the VLBA sites.

5 Phase 4: Parallel RDBE and VNDA operation

Phase 4 is the most complicated configuration during the transition from RDBE to VNDA data source. During this period, the DBS will be loaded with software that can handle both RDBE and VNDA. All software will be running at all times, but VNDA software will be mostly dormant when RDBE projects are running and vice versa.

5.1 Hardware configuration

The phase 4 hardware configuration is shown in Fig. 5.

5.1.1 VNDA

VNDA is a modular system composed of several modules connected by high-speed Ethernet. The modules are as follows:

Module ID	name	comments
D421	VNDA Producer	Dell server with COTS equipment
D430	VNDA Consumer	Dell server with COTS equipment
D440	100 Gbps switch	commodity equipment
D445	Data Buffer Server	Dell server

The VNDA Producer will sample four analog IFs in the 0-1024 MHz frequency range and multicast the raw sample streams into the 100 Gbps switch. The VNDA Consumer will channelize these raw streams and quantize to 1, 2, 4 or 8 bits per sample.

The 100 Gbps switch will likely be a “Cisco Catalyst C9500-32C” or similar device.

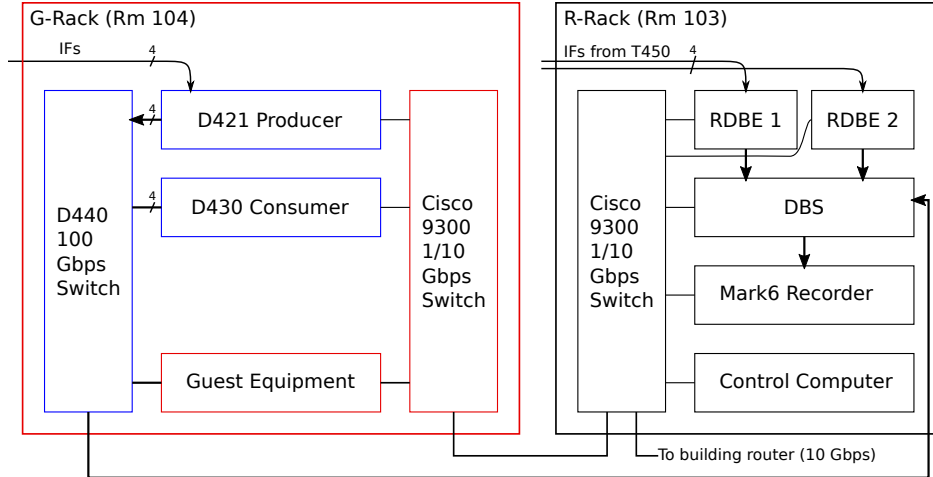


Figure 5: *Configuration of equipment in the phase 4 configuration.* Black items currently exist and will continue to be used. Blue items will be introduced by the VNDA project. Red items are to be provided by external parties. Three classes of network connectivity are indicated by line thickness. All connections to the D440 switch operate at 40 Gbps or 100 Gbps link speeds. Other connections operate at 1 Gbps or 10 Gbps.

5.2 Software configuration

The phase 4 software configuration is shown in Fig. 6. Newly introduced software components are described below.

5.2.1 vspr

`vspr` captures and formats switched power records emitted by the D430. The captured switched power data will be formatted and staged for transfer to their archive location on the computer called “elbow” in directory `/home/vlba/metadata/vnda_sp`. These calibration values are essential for amplitude calibration. The data are transferred by a program called the Remote Monitor Data Collector (RMDC).

The `rspr` source code will be used as a starting point in creating `vspr`. The following relatively minor changes will be needed:

- M&C interface will be modified to listen to VNDA configurations rather than RD BE configurations
- Change the network ports to prevent conflict with simultaneous operation of `rspr` (see table below)
- Strip out support for Mark5B-based personality
- Determine receiver band in use based on M&C query rather than inferring from channel frequency
- Support IF T_{sys} values (from the D430 or D421) as well as channel T_{sys} values (from the D430)

The network ports (and groups for multicast) currently in use by `rspr`, and those proposed for `vspr` are shown in the table below.

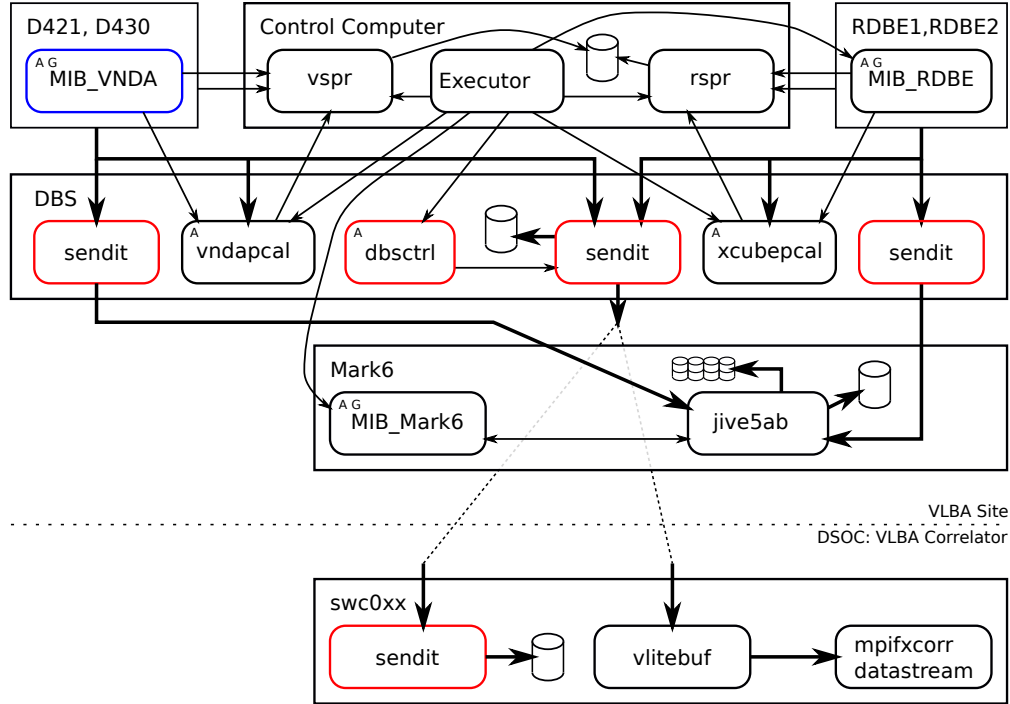


Figure 6: *Connectivity between software elements in the phase 4 configuration.* The symbology follows that used in Fig. 2. Blue items will be introduced by the VNDA project. The DBS is serving both the RDBE and VNDA systems in this phase, but at any one time only one system will have data flowing. Note that three instances of `sendit` are shown in the DBS. It is likely that the functionality of the two instances related to feeding `jive5ab` will be handled by a single instance.

program	service	port	multicast group
<code>rspr</code>	Inbound switched power from RDBE	20021	239.193.2.6
<code>rspr</code>	Calibration data on request	23098	
<code>rspr</code>	Calibration data on request (test mode)	23198	
<code>rspr</code>	Output pointing offsets to Executor	54001	239.193.3.2
<code>rspr</code>	HTTP port	55108	
<code>rspr</code>	HTTP port (test mode)	55109	
<code>vspr</code>	Inbound switched power from VNDA	20022	239.193.3.6
<code>vspr</code>	Calibration data on request	23097	
<code>vspr</code>	Calibration data on request (test mode)	23197	
<code>vspr</code>	Output pointing offsets to Executor	54001	239.193.3.2
<code>vspr</code>	HTTP port	55105	
<code>vspr</code>	HTTP port (test mode)	55106	

The “Calibration data on request” includes switched power, pulse cal tone data, and state counts. This data is primarily requested by `vlbacalserver`, which will need to be updated to look for data from `vspr`.

5.2.2 vndapcal

`vndapcal` will process the VDIF-formatted baseband data broadcast by the D430 Consumer and determine the phase and amplitude of the pulse calibration tones that are injected into the VLBA signal path at the receiver. The resultant “extracted” tones will be delivered to `vspr` for formatting and staging for transfer to

their archive location on the computer called “elbow” in directory `/home/vlba/metadata/vnda_pcal`. These calibration values aid in delay calibration.

The `xcubepcal` source code will be used as a starting point and the following changes will be made:

- M&C interface will be modified to listen to VNDA configurations rather than RDBE configurations
- Subscribe to the appropriate multicast streams from D430 rather than perform raw reads from Ethernet devices
- Strip out support for processing Mark5B data format
- Change the internal algorithms to work from complex-valued data rather than real-valued data
- Support 1, 2, 4 and 8 bits per sample; hooks for this are already in place
- Change the network ports to prevent conflict with simultaneous operation of `xcubepcal` (see table below)
- Change log file location to a location compatible with the new operating system filesystems
- Implement a VDIF-based off-line test mode; currently only Mark5B format is supported
- Generalize to support tunings that are not multiples of 250 kHz (note to developer: look at value of `nAccum`)
- Change to use the AVX2 instructions to increase compute efficiency

The multicast ports and groups currently in use by `xcubepcal`, and those proposed for `vndapcal` are shown in the table below.

program	service	port	multicast group
<code>xcubepcal</code>	pulse cal output	41222	225.2.2.2
<code>xcubepcal</code>	state counts output	41223	225.2.2.2
<code>vndapcal</code>	pulse cal output	41224	225.3.2.2
<code>vndapcal</code>	state counts output	41225	225.3.2.2

5.3 Roll-out

Phase 4 deployment will begin early in the VNDA phase 3 development stage, which is anticipated to begin around the start of FY2023 (Oct 1, 2022). Two antennas (likely FD and PT) will be outfitted with VNDA equipment. On-the-sky testing using these two units will begin shortly after deployment is complete.

Deployment to the other eight stations will begin early in VNDA phase 4, probably near the start of FY2024. This will proceed approximately two antennas at a time as VNDA production allows.

5.4 Impact to other systems

No impact to RDBE operation is expected.

5.5 Acceptance testing

All relevant testing will be covered by VNDA on-the-sky test plans.

6 Phase 5: Final VNDA-only state

In phase 5, the RDBEs are disconnected from the system, the DBS is moved from the R-Rack to the G-rack, and the Mark6 recorder is connected directly to the D440 100 Gbps switch. No new hardware or software components are introduced at this stage.

6.1 Hardware configuration

The final configuration will consist of the following addressable modules

Name	hostname	primary IP address
Control computer	<i>stn-cc</i>	10.81.subnet.11
D421 Producer	<i>stn-d421-1</i>	10.81.subnet.?
D430 Consumer	<i>stn-d430-2</i>	10.81.subnet.?
D445 DBS	<i>stn-dbs-1</i>	10.81.subnet.20
Mark6 recorder	<i>stn-mark6-1</i>	10.81.subnet.27
R-rack PDU	<i>stn-pdu</i>	10.81.subnet.252
G-rack PDU	<i>stn-?</i>	10.81.subnet.?

Some information is not yet specified for D421, D430 and the new PDU; 120 and 121 are being used as the last octets of the D421 and D430, respectively, in the test rack.

The final hardware configuration is shown in Fig. 7.

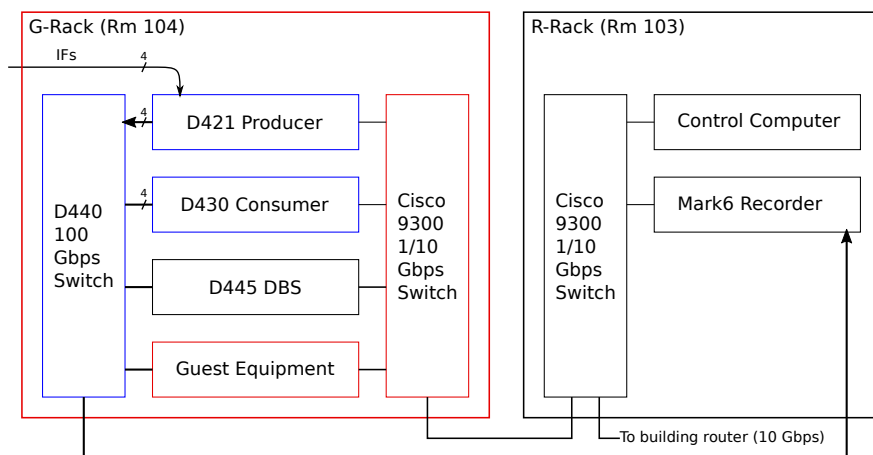


Figure 7: *Configuration of equipment in the final configuration.* Black items currently exist and will continue to be used. Blue items will be introduced by the VNDA project. Red items are to be provided by external parties. Three classes of network connectivity are indicated by line thickness. All connections to the D440 switch operate at 40 Gbps or 100 Gbps link speeds. Other connections operate at 1 Gbps or 10 Gbps.

6.2 Software configuration

The “recorder” instance of `sendit` that is dedicated to RDBEs, `rspr` and `xcubecal` will be disabled.

The final software configuration is shown in Fig. 8.

6.3 Roll-out

Transition to the final (no-RDBE) configuration will begin once scientific verification has completed and the VNDA systems are deemed worthy scientifically and further use of the RDBEs is not required. Execution of this final transition will occur on a site-by-site basis, probably over the course of a few months, with confirmation by VLBA Operations that the completed system works as expected.

6.4 Impact to other systems

After this is complete, RDBE observing will no longer be possible.

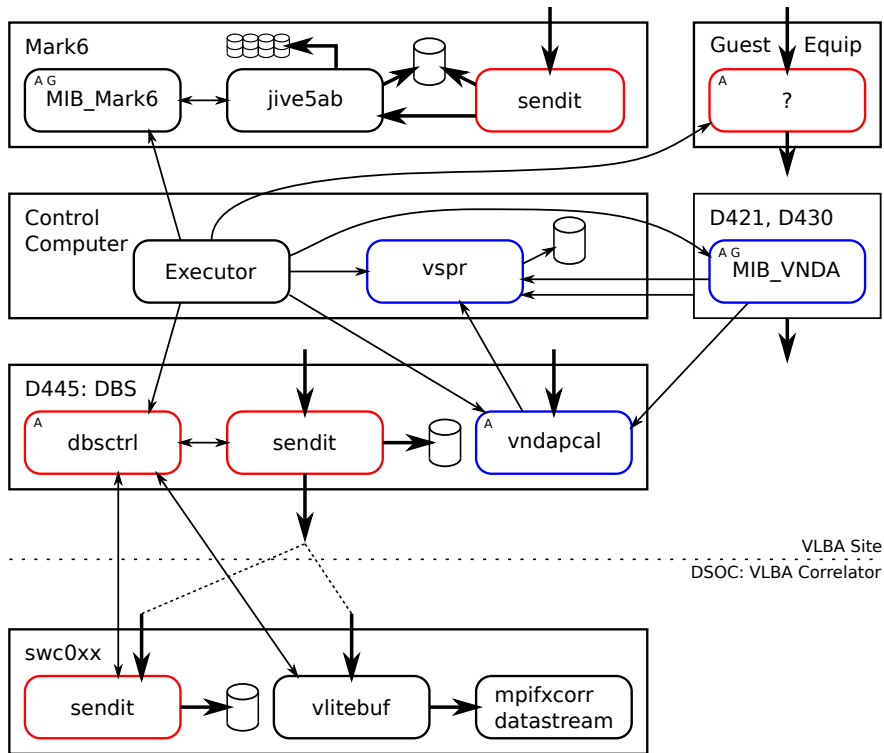


Figure 8: *Connectivity between software elements in the final configuration.* This diagram shows site software that is involved in controlling, processing or routing VDIF data. The symbology follows that used in Fig. 6. Arrows coming in from the top come through the 100 Gbps switch from the D430 (or possibly from Guest Equipment). Flow of such data is instigated by subscription to appropriate multicast addresses. The Guest Equipment can draw data either from the D421 or D430.

6.5 Acceptance testing

No fundamentally new modes of operation will be supported. Rather simple testing to ensure no data paths have been affected will be desired.

7 Software interfaces

The following sections describe the interfaces between the various software components. The interfaces described below may apply to multiple project phases. In cases where there is a change of interface between two software elements during the transition, the text will explain.

7.1 Executor to dbctrl

`dbctrl` configures the local (DBS) instance of `sendit` for transmitting VDIF data off site, or possibly to internal storage within the DBS. A separate memo, “Controlling the Data Buffer Server” is in development that describes this interface as well as the chain of interfaces upstream of that, starting from the user.

7.2 **Executor to xcubepcal**

`xcubepcal` will listen for multicast XML-formatted Observation documents from the Executor. On startup, and periodically thereafter, `xcubepcal` will make explicit requests via the Executor HTTP interface for the current state.

7.3 **Executor to vndapcal**

`vndapcal` will listen for multicast XML-formatted Observation documents from the Executor. On startup, and periodically thereafter, `vndapcal` will make explicit requests via the Executor HTTP interface for the current state.

7.4 **rspr to Executor**

`rspr` determines pointing offsets used for referenced pointing. These are multicast to port 54001 group 239.193.3.2 for the Executor to receive.

7.5 **vspr to Executor**

`vspr` will determine pointing offsets used for referenced pointing. These will be multicast to port 54001 group 239.193.3.2 for the Executor to receive. This interface will replace the existing interface used by `rspr` without change of port/group.

7.6 **dbctrl to DBS sendit**

This interfaces uses a VSI-S protocol over a TCP connection.

7.7 **dbctrl to swc0xx sendit**

This interfaces uses a VSI-S protocol over a TCP connection.

7.8 **dbctrl to vlitebuf sendit**

This interfaces uses a VSI-S protocol over a TCP connection.

7.9 **xcubepcal to rspr**

Binary blobs containing extracted pulse cal tones.

7.10 **vndapcal to vspr**

Binary blobs containing extracted pulse cal tones.

7.11 **xcubepcal to MIB_RDBE**

`xcubepcal` interrogates the configuration of the RDBE to determine the polarization and tuning of each baseband channel.

7.12 **vndapcal to MIB_VNDA**

`vndapcal` will interrogate the configuration of the RDBE to determine the polarization and tuning of each baseband channel.

7.13 D430 to DBS sendit

`sendit` running on the DBS will subscribe to the multicast addresses of the channels being created by the D430. These data streams will be routed based on commands from the Executor.

7.14 DBS sendit to swc0xx sendit

`sendit` can send VDIF-formatted baseband data to the VLBA correlator for storage. Currently there are approximately 12 filesystems set up to receive and store baseband data for near-real-time processing. RDBE data that is in Mark5B format will be converted to VDIF on-the-fly for more resilient data transfer. All transmitted VDIF packets will be 1032 bytes long (1000 bytes of data with 32 bytes of header). For the RDBE case, each 5032 byte frame will be split into five 1032 byte frames. For the VNDA case, each 8032 byte frame will be split into eight 1032 byte frames.

7.15 DBS sendit to swc0xx vlitebuf

`vlitebuf` is another possible destination for data transmitted by the `sendit` instance on the DBS. This data will be stored in a RAM buffer (up to several seconds). The data is then exposed via a FUSE filesystem, allowing the DiFX correlator to correlate in real time with appropriate buffering to hide start-up latency. The transmitted data will be of identical format to that being sent to a `sendit` instance on the VLBA correlator.

7.16 dbctrl alerts

`dbctrl` will emit alerts which will be visible on the Operator Interface.

8 Acknowledgements

Thanks to Mark Wainright for an initial read and Matt Luce for useful input and Jim Ogle for some corrections.

9 References

1. <https://vlbi.org/vlbi-standards/vdif/>
2. <https://github.com/jive-vlbi/jive5ab>
3. <https://vlbi.org/wp-content/uploads/2019/03/049.1b-Filenaming-conventions.pdf>
4. <https://safe.nrao.edu/wiki/bin/view/VLBA/Vlbanetworkports> (NRAO-internal only)