DSA1.34: Mixed-mode operations

Harro Verkouter, August 5 2009

Background

In e-VLBI the raw telescope data is volatile. It is sent from telescope directly into the correlator without ever leaving non-volatile storage. Whilst certainly most usefull, there are circumstances where "e-VLBI only" observations are either undesirable or impossible.

Notably, the following situations are quite likely to happen:

* Not all the telescopes in the observing array are connected. It is nonetheless valuable to have the real-time feedback and results of the stations that do have a connection.

* Connectivity problems. In case data link(s) go down unexpectedly during an observation.

* Unique experiments. Data which cannot be re-observed at a later time. Think of e.g. target-of-opportunity observations or a spacecraft-tracking observation.

For these usecases the "mixed-mode operation" is envisaged. In this mode the data is not just sent to the correlator in real-time but also stored on disk, either at the sending or receiving side. The full, highest resolution, dataset can be produced offline, when all diskpacks have been received at the dataprocessor at JIVE.

Feasability

The Mark5 recorders used in (e-)VLBI are equipped with a special purpose expansion card. This card features the so-called "triangle of connectivity".
The card has three bi-directional interfaces:

1) the standard VLBI data interface, VSI/H ("in"/"out")
2) the connection to the harddisks, PATA/SATA ("disk")
3) the PCI bus ("net") [because data is typically sent to the network from here]

Normally the card is programmed to perform a datatransfer from one port to

another port. In (e-)VLBI terms these transfers are historically denoted, indicating the exact transfer, as e.g. "disk2net", "net2out" etc. We use "in2disk" as an alias for "record" and "disk2out" for "play", the two classical record and playback functions. In e-VLBI the transfers are "in2net" and "net2out".

In fact, and helping the mixed-mode operation, the card has been designed such that it can send the data, as read from any single port, simultaneously
to both remaining ports. This is called "forking". To enable this feature the card has to be programmed differently and some of the commands need to be expanded. However, implementing these changes is deemed to be quite feasable.

The Mark5 hardware, in short, is in very good shape to be able to perform the mixed-mode operations we require for e-VLBI:

in2fork: data from the telescope is recorded to disk and simultaneously
sent over the network, effectively capturing data at the sending side.

net2fork: data received from the network is written to disk and
simultaneously sent to the correlator, effectively capturing the data at the
receiving end.

Changes

In order to enable the forking modes the jive5a version of the Mark5 control
(jive5a) code needed to be modified.

The "in2fork" and "net2fork" commands needed to be added.

The commands need more arguments than their single-data-transfer
counterparts:

"in2fork" requires the destination IP adress and a scanname under
which the data will be recorded to disk.

"net2fork" requires a scanname under which the data will be recorded
to disk

The expansion card in the Mark5 needed to be programmed differently,

configuring the device in "fork" mode.

Also the Correlator Control Software (CCS) had to be made aware of these new
transfer modes.

These commands exist beside their single-data-transfer counterparts and do not interfere with normal operations. By having these commands directly available, the operator can easily start or switch during an observation from one data transfer mode to another as the circumstances dictate.

Results

Jive5a and CCS were modified as described in the previous paragraph and the changes were tested during an e-VLBI testrun.

It was found that "in2fork" works reliable and does not affect e-VLBI operations at all and is well suited for use in production observations.

For "net2fork" the results were different. Technically everything works. It was found that the Mark5 performing "net2fork" looses the ability to servo. This is essential in finetuning the asynchronously incoming datastream from the network into the synchronously operating correlator. Without the ability
to servo it is basically impossible to get fringes at the correlator.