Report on FABRIC Month 7 Demonstration 'eVLBI Fringes with PC-EVN'

Editor: John Conway, Contributions from:
Metsähovi: Jan Wagner, Jouko Ritakari, Ari Mujunen and Guifré Molera
Jodrell Bank: Paul Burgess
JIVE: Nico Kruithof, Huib Jan van Langevelde and Zsolt Paragi
Onsala: John Conway, Roger Hammargren and Michael Lindqvist

21st December 2006

Abstract

Within the framework of the project plan of the FABRIC Joint Research Activity a test was conducted on 20th October 2006 to demonstrate the use of PC-EVN computers for eVLBI. The primary purpose was to demonstrate a working high bitrate (512Mbit/s) system using Commercial of the shelf (COTS) hardware. Amongst other things, PC-EVN's can be used as a test-bed for future research in eVLBI with the JRA, specifically for testing new protocols/data formats for eVLBI and new modes of data transfer (i.e. remote recording etc).

The demonstration on Oct 20th involved the radio telescopes at Onsala (Sweden) and Jodrell Bank (UK), with the Metsähovi (Finland) site as a high bit rate receiver of data and the JIVE institute in the Netherlands being used for correlation. The experiment was a great success, exceeding its objectives. In particular it (1) Demonstrated the COTS hardware worked well at data rates of at least 512MBit/s (2) Demonstrated the success of the real time Tsunami protocol, further developed at Metsähovi, at these rates (and at higher rates in pre-demonstration testing). (3) Demonstrated for the first time in Europe at high bit rate observing modes such as 'remote recording' and 'local safe data copy'. (4) Via executing a full end-to-end test from telescopes through correlation minor setup problems in the PC-EVN setup were found (buffer size allocation) that can easily be avoided in the future. (5) Finally conducting this test was very valuable in establishing personal interactions between FABRIC participants at different institutions throughout Europe. [1]

1 Introduction

The EXPReS description of work for JRA1 has as a deliverable under workpackage 1.2.3 'Broadband Integration and Test' for a demonstration in project month 7 of 'eVLBI fringes with PC-EVN'. This document describes the execution of this demonstration (known within the JRA simply as the 'Month 7 test') and what was learnt from this test.

The PC-EVN system is designed for accomplishing eVLBI using standard PC computers which with the addition of one cheap specialized interface board (built at Metsähovi) can be directly connected to a VLBI data source. Compared to the Mk5 system being used

[1]This work has received financial support under the EU FP6 Integrated Infrastructure Initiative contract number 026642, EXPReS.

for production eVLBI (and production disk recording) it is much more easily accessible for modification and experimentation.

The PC-EVN interface boards were designed several years ago and some initial tests have been done in Europe at low bit rates, however with the advent of more powerful PC's the performance of such a COTS system was expected to be greatly enhanced. The purpose of the FABRIC 'Month 7 test' was to test PC-EVN using modern hardware and so for the first time in Europe demonstrate very high bit rate eVLBI using COTS eVLBI hardware. Having established this baseline performance PC-EVN would then be available as a test-bed for experimentation within the rest of the FABRIC project. The demonstration succeeded beyond expectations surpassing the initial Month7 demonstration goals. Several small problems were encountered, fortunately fairly simple workarounds for them exist, so these will not effect future use for e-VLBI.

In Section 2 we explain the design of the demonstration. Section 3 describes the pre-demonstration tests while Section 4 describes in detail what happened during the test itself. The correlation results are presented in Section 5. Conclusions and future prospects are given in Section 6. Appendix A gives a detailed description of the Tsunami protocol used in the experiment. Appendix B gives a list of acronymns and other terms used in this document.

2 Experiment Design

Taking into account available antennas and receivers it was decided to do the test with the radio telescopes at Onsala in Sweden and Jodrell Bank in the UK operating at 6cm wavelength (or 5GHz frequency). Correlation was accomplished at JIVE in the Netherlands. Because of the need to convert the data into Mk5 format for input into the correlator fringes were not available in real time. Several different observing modes were attempted; such as recording to local computer disks at the stations and then later transfering to the correlator or direct streaming over the Internet and recording at a disk at another location ('remote recording'). Because of limitations on receiving PCs and interface cards at JIVE the highest bitrate remote recording tests were done by sending data to Metsähovi in Finland, which made use of the recently completed high bit rate connection (10Gbit/s). After being captured there the data were later transfered to JIVE at much lower bit rates ('electronic shipping') for fringe verification.

The final block schedule on Friday, 20th of October 2006 consisted of the following parts:

Part A 0900 -1000 UT. Onsala and Jodrell Bank record data to disk at 512Mbit/s, then later transfer files to JIVE for correlation. Twelve scans of four minutes were planned with one minute gaps.

Part B1 1030 - 1110. Jodrell and/or Onsala try remote recording at 256Mbit/s to disk at Metsähovi. Eight scans of four minutes were recorded with one minute gaps. Date would be simultaneously recording on disk to demonstrate a 'safe copy' mode and to prove via later file comparison that the data transfers were flawless. In addition data was transfered at low bitrate the next day to JIVE for correlation.

Part B2 1120 - 1200 As in part B1 but at 512Mbit/s.

Part C 1230 - 1330 Jodrell and/or Onsala to try remote recording at 256Mbit/s to disk at JIVE. Other station records to local disk and transfers later. Data converted at JIVE for correlation. Twelve scans of four minutes were recorded with one minute gaps.

Selection of the astronomical sources to observe and the creation of the schedule to point the radio telescopes and control the receivers was done by John Conway at Onsala helped

by Zsolt Paragi at JIVE. Control of the PC-EVN for deciding the data to be transmitted or recorded was accomplished partly by scripts distributed before the experiment and remotely over the Internet from Metsähovi with some occasional local interaction. A continuous SKYPE teleconference was conducted during the experiment to coordinate the observations.

To allow PC-EVN data recording and streaming, Jodrell Bank and Onsala were equipped by Metsähovi with PCI VSI Boards and a VSI Converter. Jodrell Bank and Onsala built their own test PC systems, and installed and customized the Metsähovi VSIB reference system based on Debian 3.1 Linux, with pre-compiled patched kernel and tools necessary for the Month 7 demonstration.

Regarding network connectivity, Jodrell Bank and Onsala stations had one dedicated test PC each, connected directly to the Internet through a 1G fibre, with appropriate routing changes made by Jodrell bank to allow access from Metsähovi. The link from Jodrell path employed 'lightpath' dedicated links to Amsterdam provided by UKLight and Surfnet. Further data transfer to Metsähovi went as non lightpath traffic over Geant and NORDUnet. Traffic from Onsala went out via normal production links via SUNET and NORDUnet to Finland and via Geant and Surfnet to JIVE. Metsähovi used two of their dedicated test PCs behind a 10G fiber. It was known that at the time of the experiment there still was a 2.5Gbit/s bottleneck in the Metsähovi link, between the university campus and the center of Helsinki. JIVE provided access to a shared-use computer behind a 1G fiber, their correlator facility and Mark5A's for the fringe checks. All five test PCs had fast RAID disks that were adequate for recording observation data at high data rates.

3 Pre-experiment network tests and preparations

A couple of days before the experiment the capacity of the backbone network was tested using the `iperf` tool, testing the PC-EVN system performance in both radio observatories at Onsala and Jodrell Bank, and finally by streaming real-time VSI-H test data from the PC-EVN systems to Metsähovi using the real-time Tsunami protocol[2], a protocol extended for real-time e-VLBI by Metsähovi. The Tsunami transfers succeeded consistently at 720 Mbit/s (one station at a time), which gave a comfortable margin above the 512 Mbit/s target.

Because everything looked just fine with a 0.0% error rate reported by Tsunami, Metsähovi decided to try a very short test to transfer at an aggregate rate of >1.4 Gigabit/s from the two stations simultaneously. Unfortunately the 2.5Gbit/s bottleneck could not tolerate this and caused a small packet loss (see Figure 1) when we tried to push through 2Gbit/s in addition to the normal traffic. The planned aggregate 1Gbit/s bitrate traffic needed to execute part B2 of the demo worked without any problems.

In addition to simply demonstrating high bite rate connectivity the pre-experiment tests produced several other specific results viz:

1) A number of small Tsunami issues were identified and dealt with during the pre-tests. This lead to Metsähovi improving the Tsunami programs for final production and then delivering these to all Month 7 participants. The Tsunami client was improved such that it could be easily controlled from an observation schedule shell-script. NTP configuration issues and other small problems found at Jodrell Bank and Onsala were detected and quickly resolved in cooperation with Jodrell Bank and Onsala staff.

2) The pre-experiment tests allowed Metsähovi to correct and fine-tune the documentation and instructions and to complete the PC-EVN data acquisition related scripts and

[2]see Appendix A - Where an introduction to Tsunami UDP protocol and Real Time VSIB transmission is described.

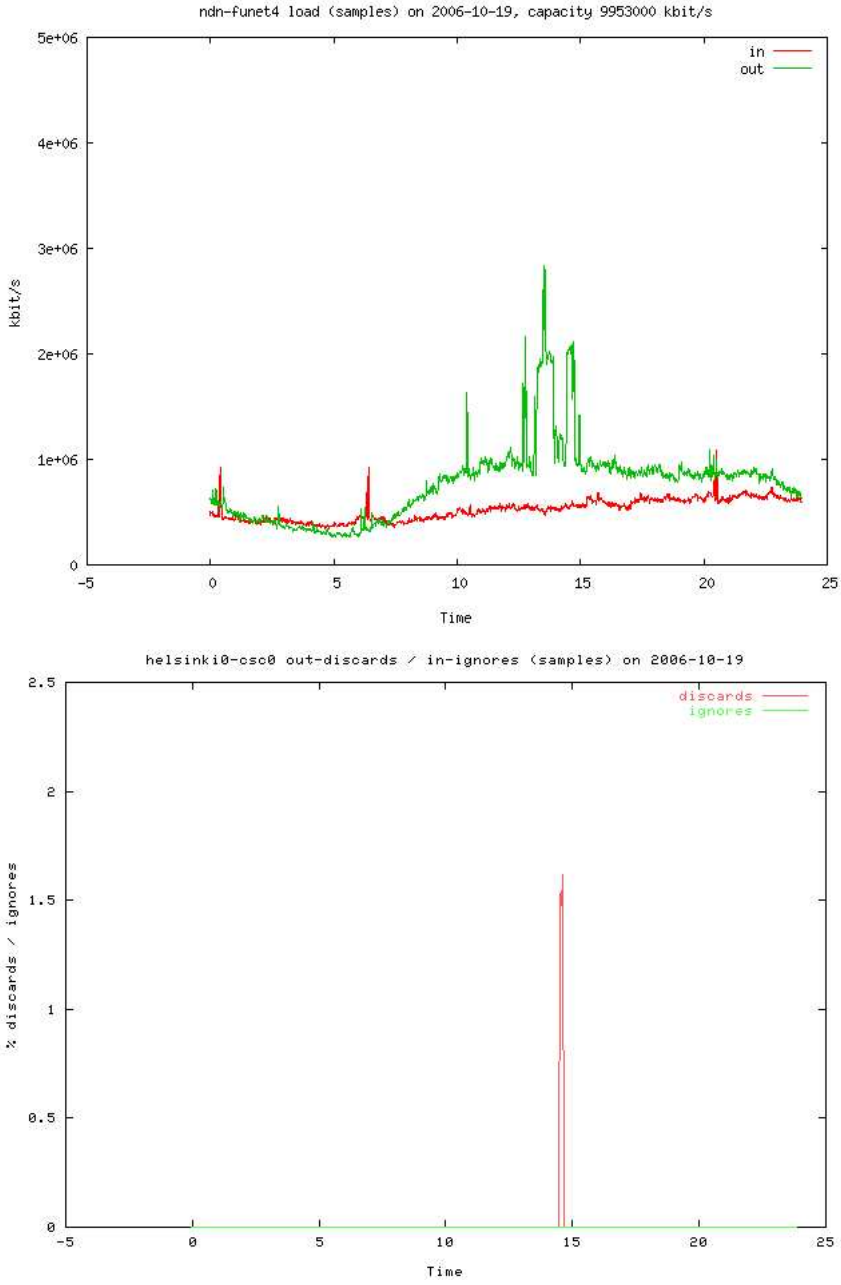

Figure 1: Top: Traffic rates via the NORDUnet link between Sweden and Finland during pre-tests on the day before the demo. The NORDUnet total output rate in green shows traffic to Metsähovi in addition to the background traffic to the rest of Finland. The horizontal axis shows hours on 19/10/2006. Time scale UT+2. Bottom: Data link quality measured as errors on the local CSC link at Helsinki, showing a small peak in discarded packets when an attempt was made to transfer to Metsähovi from two station an aggregate bit rate of 1.4Gbit/s just before 1200UT. Time scale UT+3 The traffic statistic plots are courtesy of CSC (operator of FUNET), Finland.

schedule scripts before passing them on to Onsala, Jodrell Bank and JIVE where they were tested and further customized.

3) The tests through the current Metsähovi-external bottleneck confirmed that there had been no degradation in available bandwidth compared to earlier tests, and that there was still ample headroom to achieve the 1 Gbit/s maximum total throughput needed for the Month7 demonstration.

4) As the pre-experiment tests progressed, the people at Jodrell Bank, JIVE and Onsala grew more familiar with the VSIB data capture tools, the VSIC formatting hardware and with using the Tsunami protocol. All additional PC-EVN and VSIC configurations required by the observation schedule and data transfer was worked out, tested, and found operational. This paved the way for a successful Month 7 demo.

4 Experiment on 2006-10-20

The experiment was executed according to the schedule given in Section 2, with separate parts; demonstrating respectively (A) First recording to local disk and later transfer to JIVE, then (B) remote recording (with local copy) to Metsähovi and finally (C) Attempting remote recording to JIVE. A brief summary of what happened in the various parts of the experiment is given below.

-Part A - This was terminated early after six successful scans had been locally recorded at 512Mbit/s at which point one scan was transferred to JIVE for correlation to check if everything worked. Transferring the data with Tsunami was successful, at a speed faster than the recording speed. Correlating succeeded also, but was delayed because of difficulties at JIVE in the local network and the Mark5A units.

- Part B - This was was a complete success at 256Mbit/s speeds (part B1) as we can seen in Figure 3. Typical scans at 512Mbit/s (part B2) also worked well as can be seen from the last scan remote recorded at this speed (see Figure 4) - with however some rate variations from Jodrell Bank which were later traced to a PC-EVN setup problem (see next section). However during scan 3 of part B2 the network link between Jodrell Bank and Metsähovi slowed down for a short time for some reason during scan 3 causing loss of data (see Figure 5). The link from Onsala to Metsähovi did not exhibit much rate variance at all. We suspect that on the longer Jodrell Bank-Metsähovi link other Internet traffic caused the short drops in link throughput. Future tests should show whether this suspicion is true.

The transfer time for part B2 was prolonged since the goal was to provide data integrity, instead of maintained rate with accepted data loss. Thus the scan 3 transfer extended well into the 60 second scan margin, such that scan 4 could not be started in time and was skipped. Fortunately a relatively simple protocol change can prevent this from happening in future transfers.

- Part C - This was close to getting canceled due to the unexpected problems in data transfer of part A data at JIVE from their local receiving PC to the Mark5A. Luckily JIVE managed to complete the transfer in time before Part C, and Part C could be started. It was then decided to perform Part C differently than planned. JIVE would still do real-time recording with Jodrell Bank, but additional real-time streaming would be done from Onsala to Metsähovi instead of doing Onsala local disk recording only. Unfortunately part C of the experiment failed. The real-time transfer from Jodrell Bank to JIVE met several difficulties with the Tsunami client in JIVE, despite earlier successful transfers of some Part A scans to JIVE. These problems were due to the network changes at JIVE that had to be made to transfer data scans of part A to the Mark5A's and the inability of the Tsunami protocol to handle these different settings. Meanwhile at Metsähovi, while

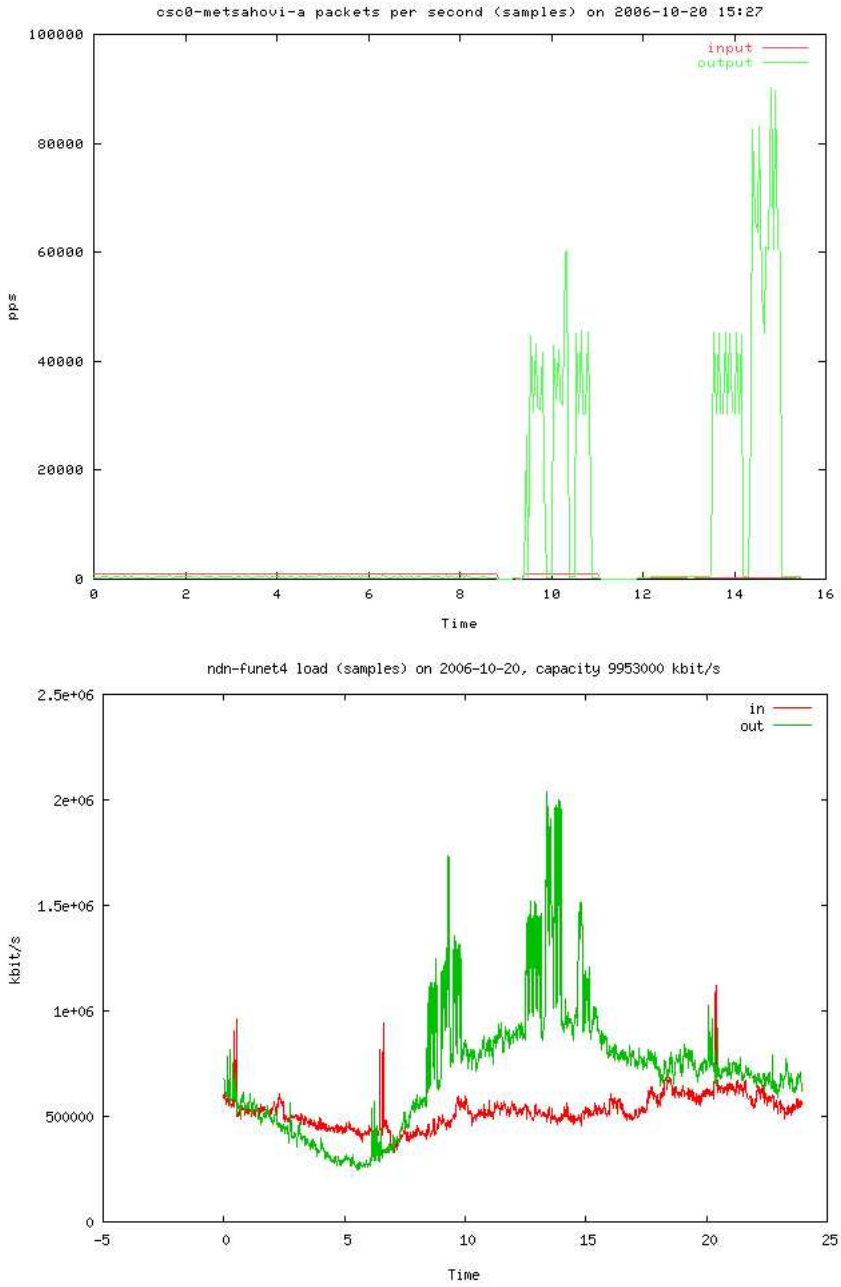

Figure 2: Top: Metsähovi input traffic (packets per second) on the experiment day 20/10/2006. It shows 3 pre-tests during the morning and the actual experiment at 256 Mbps and later at 512 Mbps. The time scale is UT+3. Bottom: FUNET traffic from (red) and to (green) Finland. It too shows the transfers of the pre-tests and experiment. The time scale is UT+2.

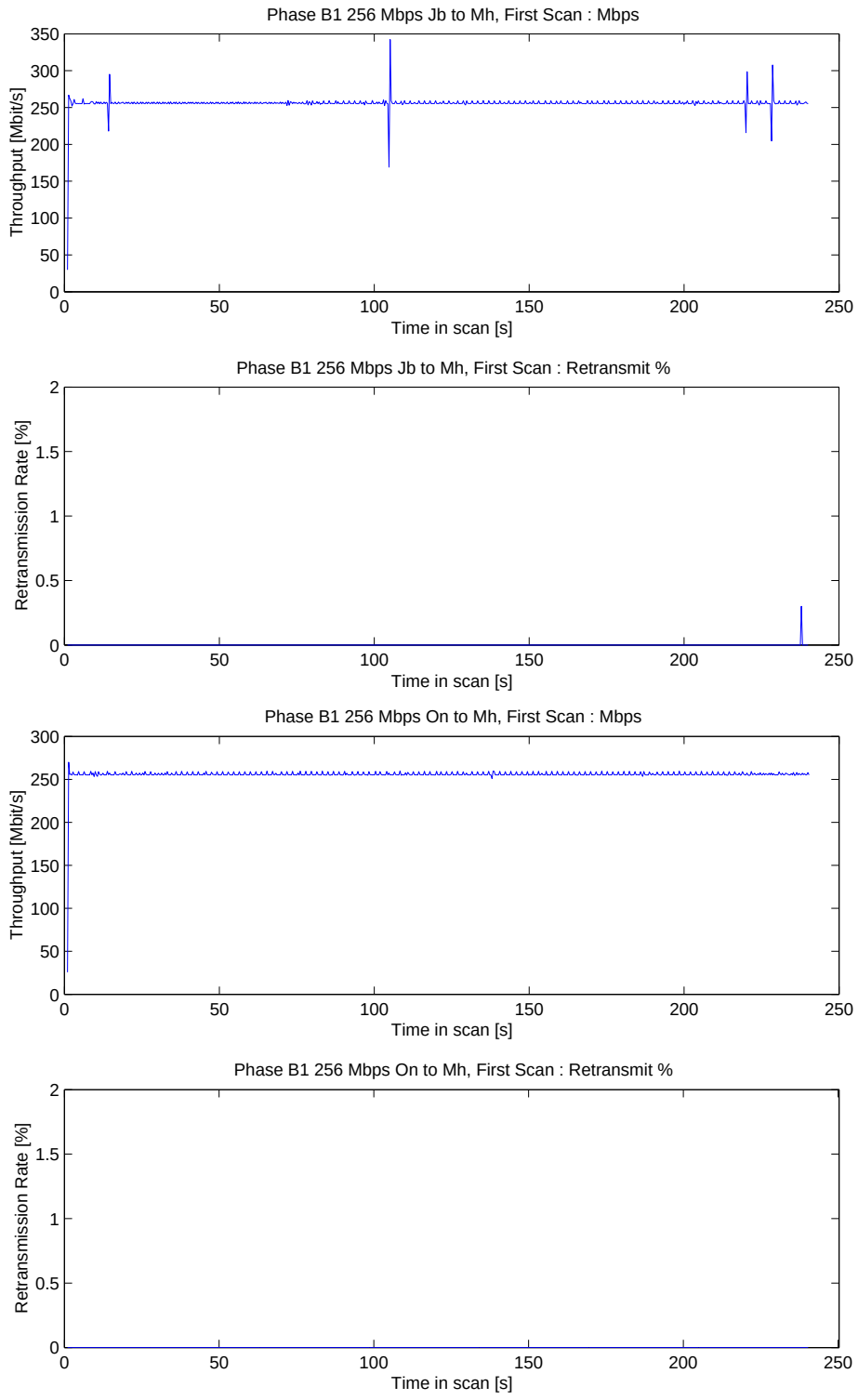

Figure 3: Transfer rate graphs generated from Tsunami logs for the first scan at 256 Mbit/s from part B1 of the experiment. The graphs show Onsala and Jodrell Bank transfers. The transfer rate is perfectly smooth during almost the entire four minutes of transmission.

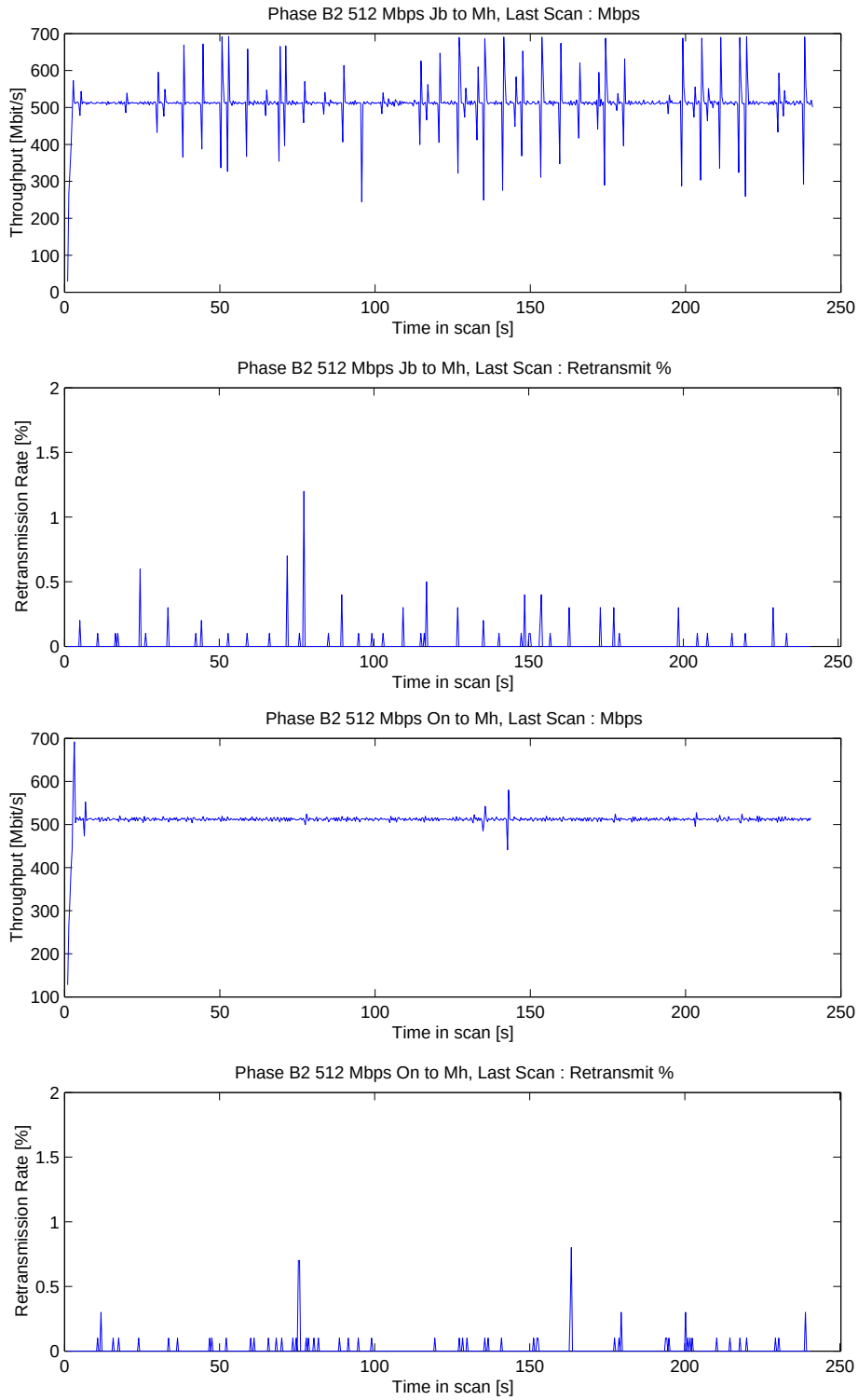

Figure 4: Transfer rate graphs generated from Tsunami logs for the last scan at 512 Mbit/s from part B2 of the experiment. The graphs show Onsala and Jodrell Bank transfers. Clearly visible is the quite smooth transfer rate for both transfers, with some rate fluctuation in the Jodrell Bank transmission.

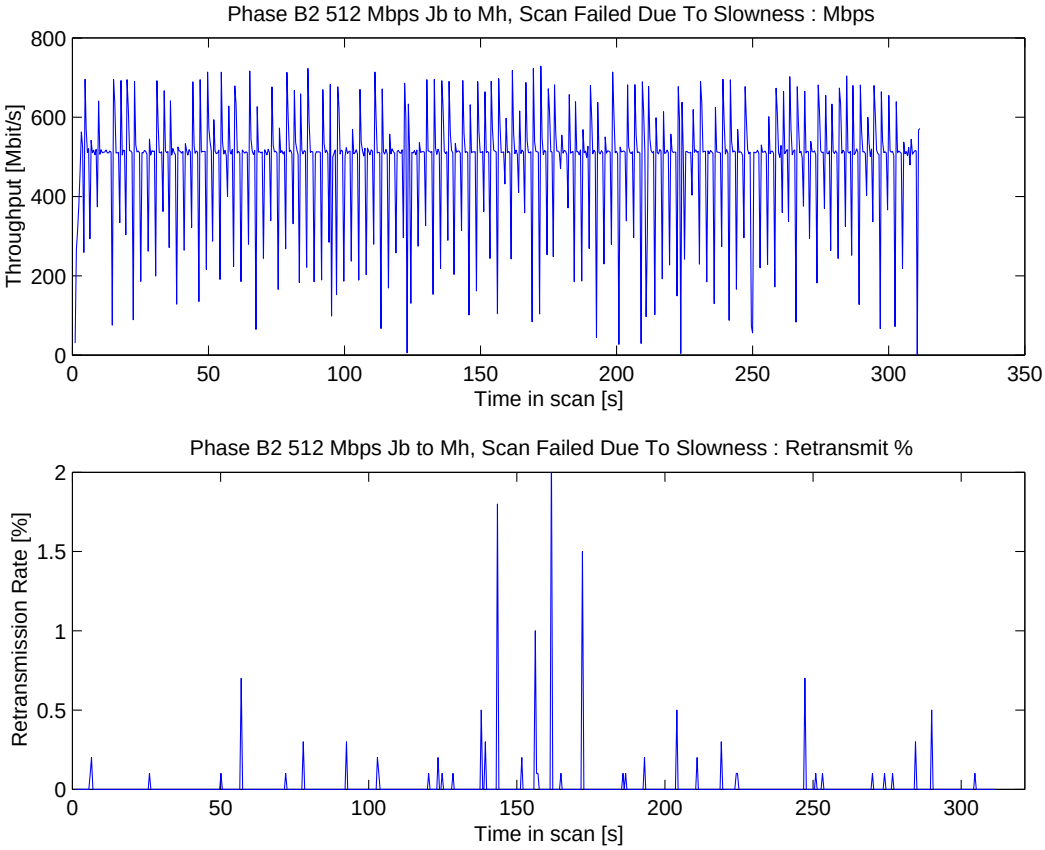

Figure 5: Link throughput graph for scan 3 at 512Mbit/s of part B2, generated from Tsunami logs, The graph shows scan 3 with 512 Mbit/s streaming from Jodrell Bank to Metsähovi. There is strong fluctuation on the transfer rate due to the smaller buffer used in the sending computer, which affected high rate transfers.

the Part C Onsala to Metsähovi streamed scans were largely successful, two scans slowed mid way. Other scans were subsequently canceled, because the corresponding scans at JIVE had failed to start at all. Continued local and online attempts to get the JIVE scans operational failed. Thus no fringes could be obtained for Part C.

5 Correlation results

The JIVE correlator successfully produced fringes for parts A and part B of the experiment. From part A the one 512Mbit/s locally recorded scan transmitted to JIVE took several hours at JIVE to be converted into the correct Mk5 input for the JIVE correlator, but fringes were available at 12:30 when part C was due to start. Data from part B from both radio telescopes was initially remotely recorded at Metsähovi but after the experiment transfered to JIVE where it was correlated. As described in the previous section part C was eventually aborted and so there were no correlation results from this part.

The correlation of scans in part B1 worked perfectly with fringes detected at full sensitivity as can be seen in Figure 6. While examining scans from the 512Mbit/s data from part A and part B2 at Jodrell Bank JIVE found short time gaps in the recorded data. The total amount of data was correct (and local and remote recorded data files identical) but due to the gaps the scan files covered a longer time than scheduled with several gigabytes of missing data within the scan and the same amount of unusable data recorded outside the scheduled time-frame. About 5% of the total record time of each scan was affected. After the experiment Metsähovi and JIVE located a buffer configuration problem at Jodrell Bank that had caused these gaps in the data and had gone unnoticed in the pre-experiment tests. The ring buffer should be configured at the correct size of 144 MB instead of the 14.4 MB used during the experiments. JIVE could still successfully get fringes for all of the B1 scans, for which the recording speed of 256Mbps was low enough not to overflow the buffer. For part A and B2 JIVE was able to produce full sensitivity fringes for the data in between the time gaps.

6 Conclusions and Future work

The objective of the Month 7 demonstration, to demonstrate fringes using commercial off-the-shelf microcomputers, has been met. The performance from the two telescopes used was comparable with the best present performance of the Mk5 production system. Causes of the temporary Internet link slowdowns in part B2 and C need further investigation. Alternatively, the simple Tsunami provision for coping with such situations should be tested. A possible new test that might be useful to conduct is to reach 1024 Mbps transfer rates. This can achieved between say Jodrell-Bank and Metsähovi by paralleling up two commodity microcomputers at both ends and using Tsunami in real-time at 512 Mbps between each microcomputer pair.

The tests with PC-EVN have has demonstrated data-rates that can easily be achieved with the Tsunami UDP protocol over the Internet with low cost hardware. The PC-EVN system and high-end commodity PCs now in place can be used as a test-bed for experimenting further with protocols. In addition it could be adapted to provide the data source for end-to-end tests in workpackage 2, scalable grid computing. This application involves time and frequency slicing the data and transmitting to multiple computation nodes. Finally the demo described in this document can be considered as a precursor for $>$ 1Gbps tests which use embedded FPGA (iBOB) technology to do the data capture and transmission instead of PCs. A final possible use might be to adapt PC-EVN to provide test-pattern input data to iBoBs which then can transmitted to the correlator allowing for verification of fringing without the need to do astronomical observations.

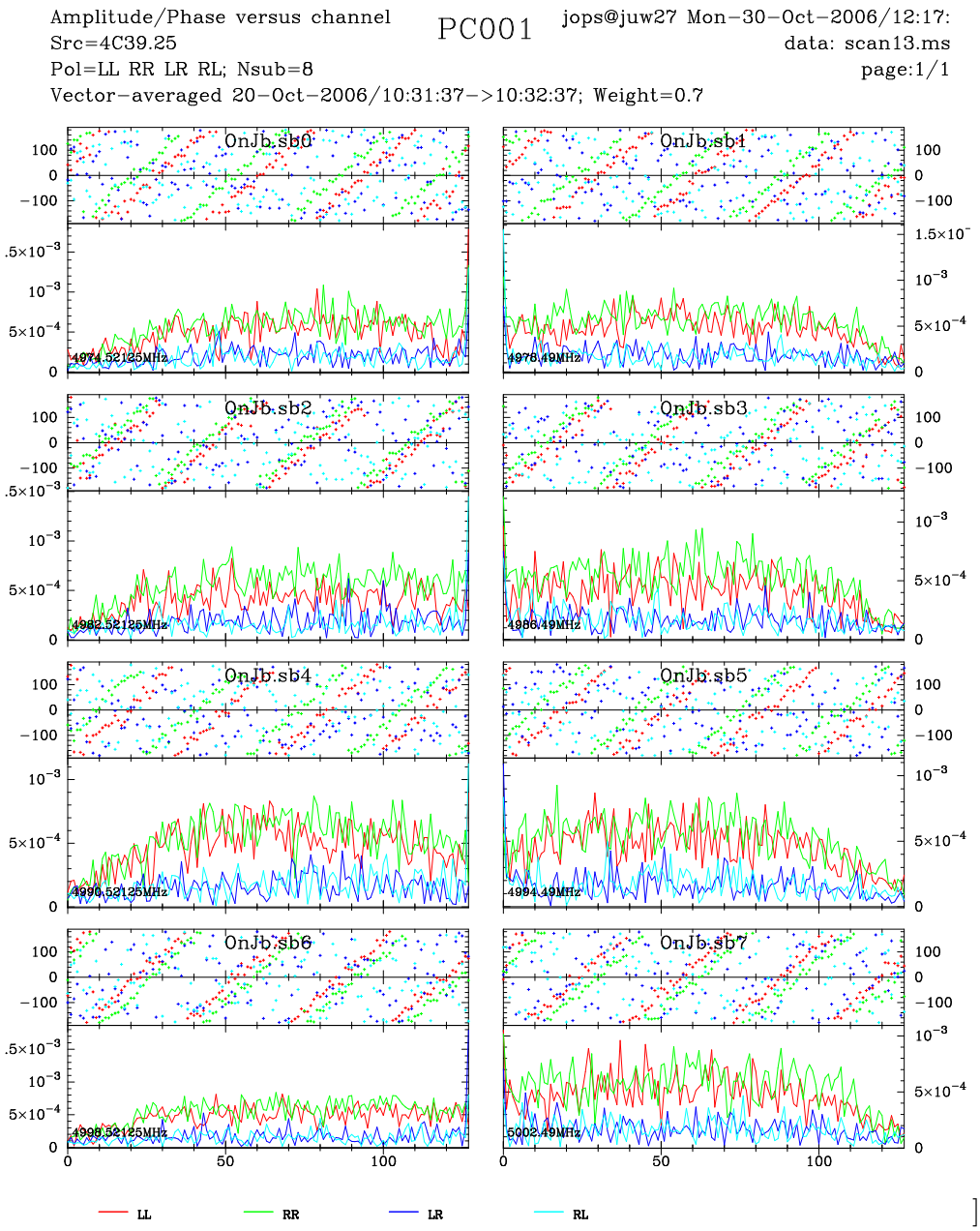

Figure 6: Plot showing correlation fringes on the Onsala to Jodrell Bank baseline for scan 13 in part B1 (at 256 Mbps). As is standard in VLBI the transmitted data from both stations is frequency filtered into sub-bands (8 in this case) before being transmitted by PC-EVN; each panel shows the fringes for one of these sub-bands. For each sub-band two data streams are transitted from each station corresponding to Left Circular polarisation and Right Circular polarisation data. In each panel the colours represent the four different correlation combinations of the data from the two stations (LL, RR, RL and LR). The bottom half of each panel shows fringe amplitude versus frequency within the sub-band, the top half shows phase versus frequency. As expected for an upolarised astronomical source the SNR of the parallel hand correlations (LL and RR shown red and green) are much larger than the cross hand correlations (LR and RL). The clear gradient of phase versus frequency in these parrallel hand correlations shows that a strong fringe has been detected.

APPENDICES

A Tsunami UDP Protocol

A.1 Brief Introduction

The Tsunami protocol was created in 2002 by the Pervasive Technology Labs of Indiana University and was released to the public as open-source under a UI Open Source License. After release, several people have improved the code and have added more functionality. Currently two branched versions of the Tsunami UDP protocol, generic and real-time, are maintained by Metsähovi Radio Observatory / Jan Wagner. Everyone interested is invited to join development at http://tsunami-udp.sf.net. A more detailed description of the Tsunami protocol and it's advantages over current TCP and other UDP based protocols can be found on the SourceForge homepage. Below is a less in-depth description of Tsunami protocol operation and the real-time extension.

A.2 How Tsunami Works

Tsunami performs a file transfer by sectioning the file into numbered blocks of usually 32kB size. Communication between the client and server applications flows over a low bandwidth TCP connection. The bulk data is transferred over UDP.

Most of the protocol intelligence is worked into the client code, it controls which blocks the server should send and when. The client specifies nearly all parameters of the transfer, such as target data rate, blocksize, target port, congestion behaviour, etc. The typical message exchange between client and server is shown in Figure 7.

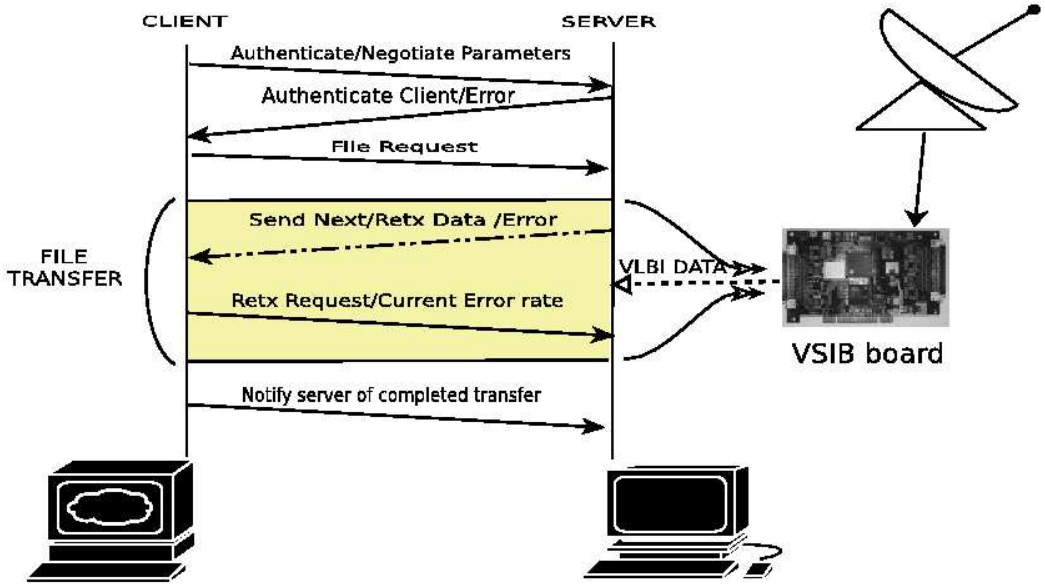

Figure 7: Design of the Real Time Tsunami UDP transfer protocol, depicting messages over the TCP communication channel. Instead of reading a file from disk, the Real Time Tsunami server captures data from the VSIB board.

Immediately after a get-file request the server begins to send out file blocks on its own, starting from the first block. It flags these blocks as "original blocks". The client

can request blocks to be sent again. These blocks are flagged as "retransmitted blocks" by the server.

When sending out blocks, to throttle the transmission rate to the rate specified by the client, the server pauses for the correct amount of time after each block before sending the next.

The client keeps track of which of the numbered blocks it has already received and which blocks are still pending. This is done by noting down the received blocks into a simple bit-field. When a block has been received, in the bit-field the bit corresponding to the received block is set to '1'.

If the block number of a block that the client receives is larger than what would be the correct and expected consecutive block, the missing intervening blocks are queued up for a pending retransmission. The retransmission "queue" is a simple sorted list of the missing block numbers. The list size is allowed to grow dynamically, to a limit. At regular intervals, the retransmission list is processed - blocks that have been received in the meantime are removed from the list, after which the list of really missing blocks is sent as a normal block transmission request to the server.

When adding a new pending retransmission to the client's list makes the list exceed a hard-coded limit, the entire transfer is re-initiated to start at the first block in the list i.e. the earliest block in the entire file that has not been successfully transferred yet. This is done by sending a special restart-transmission request to the server.

When all blocks of the file have been successfully received, the client sends a terminate-transmission request to the server.

During a file transfer, both server and client applications regularly output a summary of transfer statistics to the console window, reporting the target and actual rate, transmission error percentage, etc. These statistics may be used in e.g. Matlab to graph the characteristics of the transfer and network path.

All client file I/O is performed in a separate disk thread, with a memory ring buffer used to communicate new data from the main process to the I/O thread for writing to disk.[3]

A.3 How Real-Time Tsunami Works

The real-time version of Tsunami, created by Jouko Ritakari at the Metsähovi Radio Observatory, works in exactly the same way as the normal version. Instead of files it accesses the VSIB board developed at Metsähovi by Ari Mujunen and Jouko Ritakari.

The main difference in the real-time vs the normal Tsunami applications is that instead of accessing a "real" file like the normal Tsunami server, in the real-time version the block device "/dev/vsib" provided by the VSIB board's kernel module is opened for reading or writing. This block device supports all normal file operations, read, write, seek and ioctl.

The kernel module of the VSIB board uses a large ring buffer in main memory to store VSI data. Data read by a user from the /dev/vsib block device is actually internally read from this memory buffer. Being a ring buffer, it contains a short-term history of past VSI data. A 128MB buffer size amounts to 0.5s of 32-bit data at 512Mbit/s.

The server reads new blocks with a normal read() from the VSIB block device. The entire read data block is then sent out the normal way as an UDP packet to the client. There's no additional pause after each block since reading fresh data from the VSIB is naturally limited to the rate of data available to the VSI connector.

When the client requests blocks to be retransmitted, the server reads them out the short-term history in the ring buffer via a seek() and then normal read(). Since past data is available immediately, for these old blocks the short pause in software is implemented, to limit the transmission rate and not flood the network connection.

[3]See pseudo-code section.

Since the VSIB block device is of unlimited size, the amount of bytes to transfer is specified in the filename, together with the UTC start time for the recording. The filename contains also the option to store a local backup of sent data onto the server.

A.4 Tsunami UDP protocol Pseudo-code:

```
**Server**

 start
 while(running) {
   wait(new incoming client TCP connection)
   fork server process:
   [
     check_authenticate(MD5, "kitten");
     exchange settings and values with client;
     while(live) {
       wait(request, nonblocking)
       switch(request) {
          case no request received yet: { send next block in sequence; }
          case request_stop:            { close file, clean up; exit; }
          case request_retransmit:      { send requested blocks; }
       }
       sleep(throttling)
     }
   ]
 }


**Client**

 start, show command line
 while(running) {
    read user command;
    switch(command) {
       case command_exit:    { clean up; exit; }
       case command_set:     { edit the specified parameter; }
       case command_connect: { TCP connect to server; auth;
                               protocol version compare; send some parameters; }
       case command_get && connected:  {
           send get-file request containing all transfer parameters;
           read server response - filesize, block count;
           initialize bit array of received blocks, allocate retransmit list;
           start separate disk I/O thread;
           while (not received all blocks yet) {
              receive_UDP();
           if timeout { send retransmit request(); }
              if block not marked as received yet in the bit array {
                 pass block to I/O thread for later writing to disk;
                 if block nr > expected block { add intermediate blocks
                    to retransmit list; }
              }
              if it is time {
```

```
                process retransmit list, send assembled request_retransmit to server;
                send updated statistics to server, print to screen;
            }
        }
        send request_stop;
        sync with disk I/O, finalize, clean up;
    }
    case command_help:    { display available commands etc; }
  }
}
```

B Acronyms

COTS	Commercial Of The Shelf
CSC	Computer Science Centre (operator of Finnish FUNET)
eVLBI	Electronic VLBI
FABRIC	Future Arrays of Broadband Radio-telescopes on Internet Computing
FUNET	Finnish University and Research Network
I / 0	Input / Output
JIVE	Joint Institute for VLBI in Europe
Mk5	Mark V
NORDUnet	Nordic University Network
NTP	Network Time Protocol
PC-EVN	Personal Computer - Electronic VLBI Network
PCI	Peripheral Component Interconnect
SUNET	Swedish University Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
VSIB	VLBI Standard Interface Board
VSIC	VLBI Standard Interface Converter
VLBI	Very Long Baseline Interferometry