Future Generation Computer Systems 26 (2010) 120–127

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

VLBI_UDP: An application for transporting VLBI data using the UDP protocol

Simon Casey [a,*], Richard Hughes-Jones [a], Ralph Spencer [a], Matthew Strong [a], Paul Burgess [a], Arpad Szomoru [b], Colin Greenwood [c]

[a] School of Physics and Astronomy, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK

[b] Joint Institute for VLBI in Europe (JIVE), Dwingeloo, Netherlands

[c] ASTRON, Oude Hoogeveensedijk 4, 7991 PD Dwingeloo, Netherlands

ARTICLE INFO

Article history:
Received 12 July 2007
Received in revised form
3 October 2008
Accepted 7 October 2008
Available online 25 October 2008

Keywords:
Astronomy
Data communications
Fault tolerance
Internet
Network communication
Network monitoring

ABSTRACT

This paper describes the VLBI_UDP application, which has been designed to transport astronomical data recorded via a technique known as Very Long Baseline Interferometry (VLBI), using the UDP protocol. The design of the VLBI_UDP application is discussed together with modifications that provide additional features such as file access and packet dropping. Network tests of the application show that the application is stable and correctly deals with packets lost on the network. Correlation tests using data with deliberately dropped packets show that the correlation process is resilient under data losses approaching 20% and that the resulting correlation amplitude decreases as 1.2 times the packet-loss rate. The results also indicate that while the correlator is able to cope with at least 2 consecutive missing headers, it needs to see at least 2 correct headers, otherwise synchronisation is lost.

1. Introduction to e-VLBI

VLBI is a technique [1] used in radio astronomy for achieving sub-milli-arcsecond resolution images by combining simultaneous observations made with telescopes situated hundreds or thousands of kilometres apart. Signals from the telescopes are sent to a high performance computer which multiplies the signals from each pair of antennas to extract the information from the noise present in the received signal. A Fourier Transform of the processed information produces an image of the radio source with real and imaginary components. This process is known as correlation, hence the computer is known as the correlator. For n antennas, there are $\frac{n(n-1)}{2}$ possible signal pairs (baselines) which are calculated simultaneously.

Since VLBI antennas can be situated across continents, until now it has not been feasible to have real-time links between them and the correlator, and so traditionally the data have been recorded to tapes or disks then shipped to a correlator for off-line processing. e-VLBI replaces the physical movement of tapes and disks, by instead transferring the data from the antennas to the correlator over high-speed computer networks and the academic Internet. PC-based systems called Mk5A [2] are used to interface the telescope and correlator front end to the network. This produces a real-time interferometer, resulting in much quicker access to data, higher resilience by being able to monitor the correlator output and being able to act on any problem, and the enabling of Target of Opportunity (ToO) observations. Currently, VLBI data rates of 256 Mbit/s per antenna are routinely achievable using the Transmission Control Protocol (TCP) and occasionally rates of 512 Mbit/s are possible [1,3]. An alternative to TCP is User Datagram Protocol (UDP), as used by a new software application, VLBI_UDP.

This paper discusses the suitability of using UDP/IP as a transport protocol for e-VLBI, and then outlines the structure of VLBI data and how the correlator receives this data. This is followed by presentation of the architecture and implementation of the VLBI_UDP application which provides efficient data transport as well as allowing portions of data to be lost before transmission, simulating data loss over a real network. Next a section describes the use of VLBI_UDP to prepare VLBI data with selected data packets to be dropped, allowing investigation of the behaviour of the VLBI correlator under data-loss conditions. Finally the results of network tests and investigations of the effect of varying rates of packet loss on the correlator output are presented.

2. The case for using UDP for e-VLBI

TCP is a window-based protocol designed to ensure that all the data sent arrive uncorrupted at its destination and in the correct

* Corresponding author. Tel.: +44 01477 572658.
E-mail address: scasey@jb.man.ac.uk (S. Casey).

0167-739X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2008.10.006

order. To do this, it uses acknowledgements of the data received to detect lost data which is then retransmitted. TCP also implements a congestion control algorithm which adjusts the transmission rate if TCP perceives the network as being congested. The standard TCP congestion control algorithm reduces the transmission rate by half if any packet loss is detected. The higher the round trip time (RTT), which is proportional to the physical network distance, the longer it takes TCP to recover back to its previous rate after a packet-loss event [4]. These factors give rise to concern that TCP may have difficulty in delivering the data in a timely manner as required for e-VLBI [5] if there is any loss on the network.

Since the correlator processes data from pairs of antennas and all pairs are processed simultaneously, data from all antennas must arrive synchronised in time at the correlator. There are both software and hardware buffers in the Mark5A systems and the Station Units, that form the input to the correlator, which can compensate for a reduced transmission rate for short periods of time, but extended slow periods would mean the buffers run empty. Should 1 antenna from a group of *n* fall behind the others, this would result in the temporary loss of *n* baselines. Indeed, should TCP detect a lost data segment then it will halve the transmission rate causing the receiving buffer to empty faster than it is being filled. If TCP is unable to recover the transmission rate before the receiving buffer has emptied, then that antenna will drop out from the correlation.

UDP, on the other hand works on a best effort basis, leaving it up to the application to monitor for any congestion, control transmission rates and detect any data which failed to arrive. If the application were to send data at a constant rate, UDP would ensure that the reception rate is kept constant, albeit with small portions of data missing corresponding to any lost or corrupted packets. The requirements for moving VLBI data are unusual compared to those usually found in bulk file replication or transfers. The data from each telescope consists of white noise, so the loss of an individual packet is not important provided that the loss is known and the data is replaced with artificial random data prior to being fed into the correlator. Loss of data of course results in a corresponding decrease in the final signal to noise ratio, but the loss of a few percent can be tolerated. The correlator was originally designed to be fed with data from magnetic tapes and so is resilient to a certain amount of data loss [6]. The e-VLBI requirements for high bandwidth, long distance data flows, and timely arrival of data together with resilience to packet loss make UDP a likely choice of transport protocol.

There are several other UDP-based protocols in use, such as Tsunami UDP, RUDP and RTP/UDP. Both Tsunami and RUDP ensure that data are reproduced faithfully at the receiver, retransmitting any missing packets. Whilst their algorithms for doing this allow for data rates higher than that possible with a TCP transmission, if a link is already being run at for example 99% capacity, then to retransmit a packet which failed to arrive will most likely be at the detriment of other data, and may even arrive too late for it to be useful anyway. This feature makes these protocols not entirely suitable for real-time e-VLBI. RTP/UDP does not retransmit missing data, instead it is up to the application to decide if data should be resent. What makes RTP/UDP unsuitable for real-time e-VLBI however, is that it was designed for real-time audio and/or video, where data rates may be several Mbit/s at most as opposed to e-VLBI with rates currently of up to 1 Gbit/s. As a consequence, the 32 bit timestamp field and 16 bit sequence number are insufficient for e-VLBI; at 1 Gbit/s the sequence number would wrap at least once per second assuming 1400 bytes per packet. Owing to the negative aspects of these protocols when dealing with e-VLBI, it was therefore decided that a custom UDP application header should be created for VLBI_UDP.

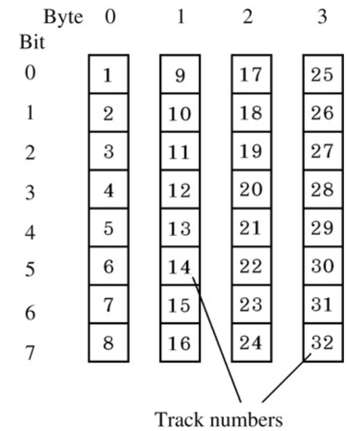

Fig. 1. Visual layout of VLBI data tracks within PC file.

3. Details of VLBI data and the correlator input

This section outlines the data format used by the Mark5A data recorders, and the way this data is presented to the correlator. This background helps to explain the tests that were run to examine how the correlator would cope with lost data.

3.1. Mark5A data format

The data fed into the Mark5A PCs at each telescope are divided into "tracks", each corresponding to a particular physical measurement domain comprising polarisation and frequency band. These tracks come from a Mark4 tape formatter which frames the raw signals and adds headers and trailers with time stamps and other information to each track. The Mark5A has a variety of modes [7] which affect exactly how the data stream appears when written to a standard PC file. The data used in these tests were recorded at a rate of 256 Mbit/s, 32 tracks with 1-bit sampling. Upon examining the recorded file, it became apparent that the VLBI data tracks were bit-interleaved as in Fig. 1, such that 2 subsequent bits from any track would be separated by 31 bits in the PC file. Each track is framed in 2500 byte segments, and so 32 tracks give a 'combined frame' size of 80,000 bytes. This can then be thought of as 79,360 bytes of VLBI data sandwiched between a 384 byte header and a 256 byte trailer.

3.2. Keeping Station Units in synchronisation

The Station Units (SUs) sit between the Mark5A PCs that receive the data from the network and the correlator. They are a relic from the days of tape and, were designed to regulate the incoming data stream to keep all the streams aligned in time as well as checking that the data has valid headers. Since the Mark5A was designed as a direct replacement for the tape drives, the SUs are still required. To ensure all the data streams are synchronised in time, the SUs read the headers from each frame, perform various checks to ensure that the headers are valid, and then read the time stamp to ensure synchronisation. Since the data frames are a constant 2500 bytes long, the SUs know exactly where in the stream the next header should be located. If the SU does not find a valid header where it expects to see one, then it is able to flywheel (or skip) over that frame and check where the subsequent header should be. It was unknown how the SUs would behave when there were multiple headers missing, giving rise to the investigations where only the VLBI headers were dropped.