PROCEEDINGS OF SCIENCE

e-VLBI Networking Tricks

Paul Boven*, **for the EXPReS team**
Joint Institute for VLBI in Europe (JIVE)
E-mail: boven@jive.nl

Real-time e-VLBI is made possible by having access to long-distance, high-performance networks - straight from the radio telescopes into a correlator facility. This article discusses what we've learned in the EXPReS project about how to make the best use of these networks. Topics include the importance of packet spacing, the use of multicast and trunked paths.

Science and Technology of Long Baseline Real-Time Interferometry:
The 8th International e-VLBI Workshop, EXPReS09
June 22 - 26 2009
Madrid, Spain

*Speaker.

PoS(EXPReS09)036

1. Network overview

The EXPReS project, with much support from the involved NRENs and DANTE, has built a global network of connections from radio telescopes to the JIVE correlator. We use a mix of different networking technologies: SDH-based lightpaths (LP), VLAN-based paths, routing through interconnected research networks, and in one case, CWDM over a dark-fiber connection. Table 1 and figure 1 give an overview of the current state of the network.

Station	**Type**	**Speed**	**RTT**	**Comments**
Sheshan	LP	622 Mb/s	354 ms	via Canarie
	Routed	500 Mb/s	180 ms	via TEIN2
Urumqi	LP	622 Mb/s	374 ms	Temporary, same LP as Seshan
ATNF	LP	1 Gb/s	343 ms	ATCA, Mopra and/or Parkes
Kashima	Routed	512 Mb/s	288 ms	
Arecibo	VLAN	512 Mb/s	154 ms	Only at certain times of day
TIGO	Routed	95 Mb/s	150 ms	Bandwith must be arranged prior
Westford	Routed	512 Mb/s	92 ms	
Hartebeesthoek	Routed	64 Mb/s		Temporary capacity on SAT-3
Yebes	Routed	990 Mb/s	42.1 ms	
Torun	Routed	>1024 Mb/s	34.9 ms	LP and routed
	LP	1 Gb/s		both available
Onsala	VLAN	1.5 Gb/s	34.2 ms	NorduNet TSS
Metsahovi	Routed	>1024 Mb/s	32.7 ms	
Medicina	LP	1 Gb/s	29.7 ms	
Jodrell Bank	2x LP	2x 1 Gb/s	18.s ms	Also to MERLIN stations
Effelsberg	VLAN	10 Gb/s	13.5 ms	Shared with e-LOFAR
WSRT	CWDM	2x 1 Gb/s	0.57 ms	Dark Fiber

Table 1: Network connections to JIVE

2. Packet timing

e-VLBI data are generally transported by several different networks to reach the correlator: a local area network at the telescope, a long distance path (e.g. a lightpath) and then the local network at JIVE. Problems occur when any section of the path has a lower bandwidth than the interface speed of the sending Mark5 at the telescope, even if this bottleneck bandwidth is sufficient to sustain the requested average data rate. Some examples of such bottlenecks that we encountered:

- Sheshan: 1 Gb/s interface, bottleneck: 622 Mb/s lightpath from Hong-Kong to JIVE.
- Onsala: 10 Gb/s interface, bottleneck: 1.5 Gb/s rate policer on the NorduNet TSS cloud.
- Torun: 10 Gb/s interface, bottleneck: the 1 Gb/s connection from the JIVE switch to a JIVE Mark5 (most Mark5 at JIVE are now connected at 10 Gb/s).
- ATNF: Sharing a 1 Gb/s lightpath with two telescopes each transmitting at 256 Mb/s from a 1 Gb/s interface

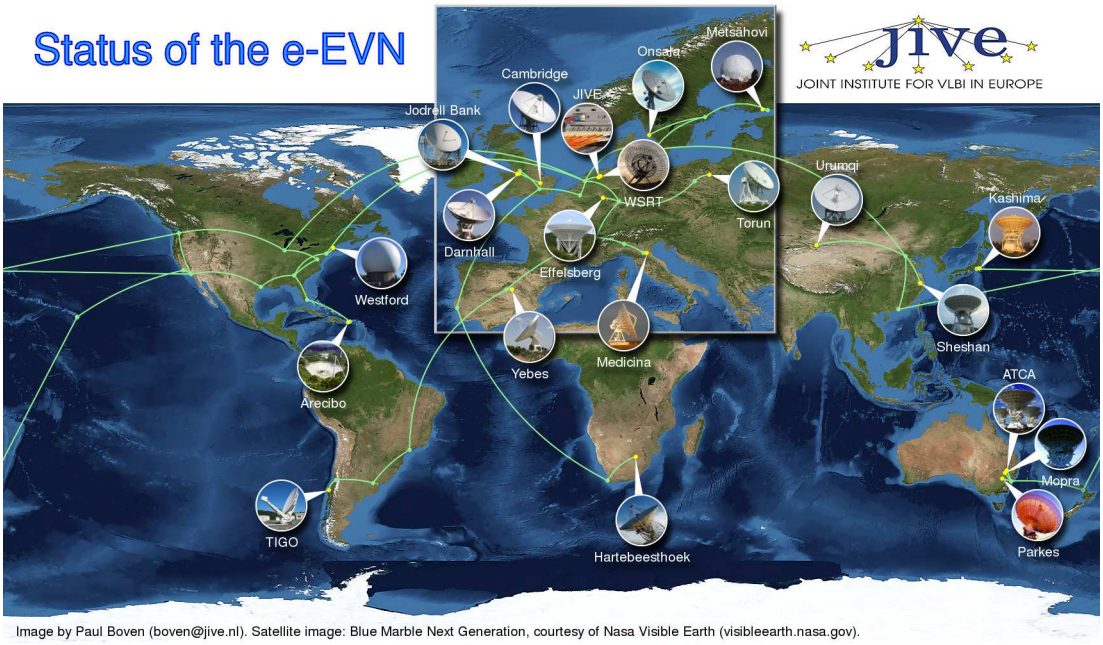

Figure 1: EXPReS e-VLBI Network overview

A naïve implementation of UDP e-VLBI transfer would simply send out all the available data as packets, and then wait for more data to become available. This approach would cause a rather bursty traffic pattern: once the sending thread has relinquished the CPU, it will only become scheduled again at the next kernel scheduling interrupt. With a standard Linux 2.6 kernel the scheduler interrupt only fires every 4 ms. When sending e-VLBI data at a rate of 512 Mb/s (528 Mb/s including headers), this hypothetical naïve implementation would send data at 1 Gb/s for slightly over 2 ms, followed by almost 2 ms of silence. Each of these bursts will consist of 2.1 MB of data, or 1474 packets of 1500 bytes. At a choke point, e.g. the start of a 622 Mb/s lightpath, the data can only be sent onwards at the lower rate. The device at the choke point therefore must have sufficient memory to store the incoming flood. Often it doesn't, causing packets to be dropped even though the average data rate is lower than the link capacity.

Ideally, one would want to add a small delay between each transmitted network packet to have a constant data rate on both short and long timescales. The Linux scheduling interrupt is too coarse for this, and the jive5a [1] application implements these delays by executing a calibrated, tight delay loop on one of the CPUs of a Mark5. This indeed gives very fine-grained control of the 'inter-packet delay' (IPD) and eliminates the packet loss due to burstiness. The drawback of the current delay-loop implementation is that it fully utilizes a CPU core to generate the delays. We have also used the real-time features of recent Linux kernels, especially the high-resolution timers, to generate the required short delays without wasting so much CPU resources. In our first tests, we typically observed that the CPU load for the thread that transmits these packets went from 99% to only 5% on a standard Debian Lenny (2.6.26) kernel.

3. Flooding

To learn behind which port a specific device (Ethernet hardware address) is connected, a network switch will initially flood packets for unknown destinations to all the member ports of a VLAN. Once the device replies, the switch knows behind which port it is and can update its address table. Subsequent packets will then only be forwarded through the proper port. A switch however will keep flooding the traffic to all ports if it never learns behind which port a destination is connected; this can happen because the e-VLBI traffic itself is uni-directional. Another cause for flooding is when a network port on the switch goes down, e.g. because the connected Mark5 crashes and reboots: This will cause most switches to immediately remove the destination from their forwarding table. If the remote side continues to send traffic, it will be flooded to the remaining ports. This is a problem especially when using e-VLBI at 512 Mb/s: a Mark5 at JIVE that is already receiving 512 Mb/s of data from a particular telescope will then be sent another 512 Mb/s of flooded traffic. The total amount of traffic will exceed the 1 Gb/s capacity of the connection to the Mark5, causing severe congestion and packet loss. This problem frequently plagued our first e-VLBI runs at 512 Mb/s but was solved by creating a separate VLAN and IP-space (/30) for each of the Mark5 servers at JIVE, at the cost of using a much larger part of our IP assignment for the Mark5 servers.

4. Trunking

The highest bandwidth supported by the current EVN correlator [2] is 1024 Mb/s, which is slightly more than can be carried by a 1 Gb/s Ethernet connection. Trunking is a cheap way to overcome this 1 Gb/s speed-bump: by installing a second 1 Gb/s interface in the Mark5, each network interface will only have to carry 512 Mb/s of traffic (plus headers and other overhead). The complete network path to JIVE will of course have to be doubled, but this is generally much cheaper than upgrading to the next available Ethernet speed, which is 10 Gb/s.

The standardized method of configuring trunks is called LACP (Link Aggregation Control Protocol), which is supported by most networking equipment. This standard specifies that network traffic belonging to a single 'stream' must only be forwarded over only one of the link-members, to prevent packet re-ordering within such a stream, as that has a marked negative effect on TCP performance. The link on which to send out a packet on a trunk is generally decided by using a fixed hash of sender and destination MAC-address, TCP/IP address and/or UDP port. This makes LACP-based trunks unsuitable for e-VLBI data which is sent as a single UDP stream, as this would use only one of the available links without an increase in usable bandwidth.

The Linux kernel also supports a round-robin trunking mode where packets are simply distributed amongst the trunk members in an alternating or sequential way. This ensures equal traffic levels on the link members. The Linux 'ifenslave' command is used to bind two network interfaces together like this. As most network equipment along the way will only support LACP-based trunks, it is important to create two completely independent network paths that will each carry half the traffic. These two paths will terminate at the central JIVE switch/router. Although the switch has no problem receiving traffic distributed in this way over two links, it too only supports LACP-type distribution as a sender. This precludes the use of another trunk between the JIVE switch and the

receiving Mark5 at JIVE, which is why CX-4-based 10 Gb/s Ethernet is used for the short distance between the switch and the receiving Mark5s at JIVE.

Inter-packet spacing is important when trunking is used: without it, the network cards would simultaneously send bursts of back-to-back packets. The order in which these streams get reassembled at the end of the trunk then becomes indeterminate. Instead of delivering the packets in-order, they can also arrive as $2,1,4,3,\cdots$ causing half the packets to be discarded as being out-of-order. Inserting the proper delay between the packets ensures that the streams will be reassembled in the correct order.

5. Multicast

The MERLIN (Multi Element Radio Linked Interferometer Network) in the UK consists of 7 telescopes: two at Jodrell Bank, and 5 which are connected to the Jodrell Bank facility by 128 Mb/s microwave links. Including multiple MERLIN outstations in EVN VLBI observations adds short spacings to the array, which increases the sensitivity to large-scale structures in the reconstructed image. As the throughput of these links is limited, and the outstations are all synchronized to the same clock, it turns out to be possible to record the data for up to 4 of these Merlin outstations simultaneously on a single Mark5 recorder at Jodrell Bank. In disk-based VLBI, the disk packs are then sent to JIVE, where they are duplicated and replayed in parallel on different Mark5 servers, with each Station Unit (SU) applying the appropriate delay model to the data tracks from a particular telescope.

In e-VLBI, duplicating the transmitted data to make it available to multiple Mark5s and SUs at JIVE has to happen in real-time. IP Multicast [3] is a special IP addressing mode where multiple receivers can subscribe to a stream of data. It uses unacknowledged UDP datagrams, just like our e-VLBI application. One of the important features of Multicast is that a stream is only sent over those links that lead to subscribers of that particular stream, and that the routers or switches along the way take care of duplicating the packets where needed. Because of the high data rates of up to 512 Mb/s, the switch/router does need to support Multicast routing in hardware. This way it becomes possible to send a stream just once over a lightpath from Jodrell Bank to JIVE, and the central e-VLBI switch/router at JIVE then sends copies of this data stream to any Mark5 at JIVE that subscribes to it. This mode of operation has been dubbed 'Merlincast' and is regularly used in production e-VLBI. Even recorded VLBI sessions that include multiple Merlin dishes are now played back and duplicated locally over the JIVE network, as this both saves time and doesn't require extra disk packs for copying the Merlin station data.

6. The Elliptical Robin

From Jodrell Bank there are two 1 Gb/s lightpaths to JIVE. Using both paths connected to a single Mark5 and using trunking, data from Jodrell Bank (either the Lovell or Mark II telescope) can be transported to JIVE at the full 1024 Mb/s. It is also possible to use one lightpath for 512 Mb/s of data from a Jodrell Bank telescope, and use the other lightpath for the second Mark5, sending data from one or more of the MERLIN outstations. But with only two lightpaths it isn't normally possible to have both the full bandwidth from Jodrell Bank, and the Merlincast data from the

outstations all transmitted at the same time: The Jodrell Bank data would already put 512 Mb/s on each lightpath, not leaving enough room on either to transport the additional 512 Mb/s Merlincast data. However, an 8 line change to the Debian Etch kernel for the Jodrell Bank Mark5 makes it possible to 'skew' the distribution of packets over both the links: the modified Linux bonding driver sends several packets over the primary interface before transmitting one over the secondary interface. This modification of the round-robin driver has been dubbed the 'Elliptical Robin'. With a 4:1 packet distribution, one link will carry about 820 Mb/s, and the other only 204 Mb/s to make up a full 1024 Mb/s. This leaves sufficient space on the secondary link to add 512 Mb/s of Merlincast traffic.

References

[1] Verkouter, H. *jive5a software packages*, `http://www.jive.nl/~verkout/evlbi`

[2] Schilizzi, R.T. e.a., *The EVN-MarkIV VLBI Data Processor*, Experimental Astronomy, Vol 12, Nr 1, 2001, pp. 49-67

[3] Deering, S. *Host extensions for IP multicasting*, RFC-1112 `http://www.ietf.org/rfc/rfc1112.txt`