Mark 6

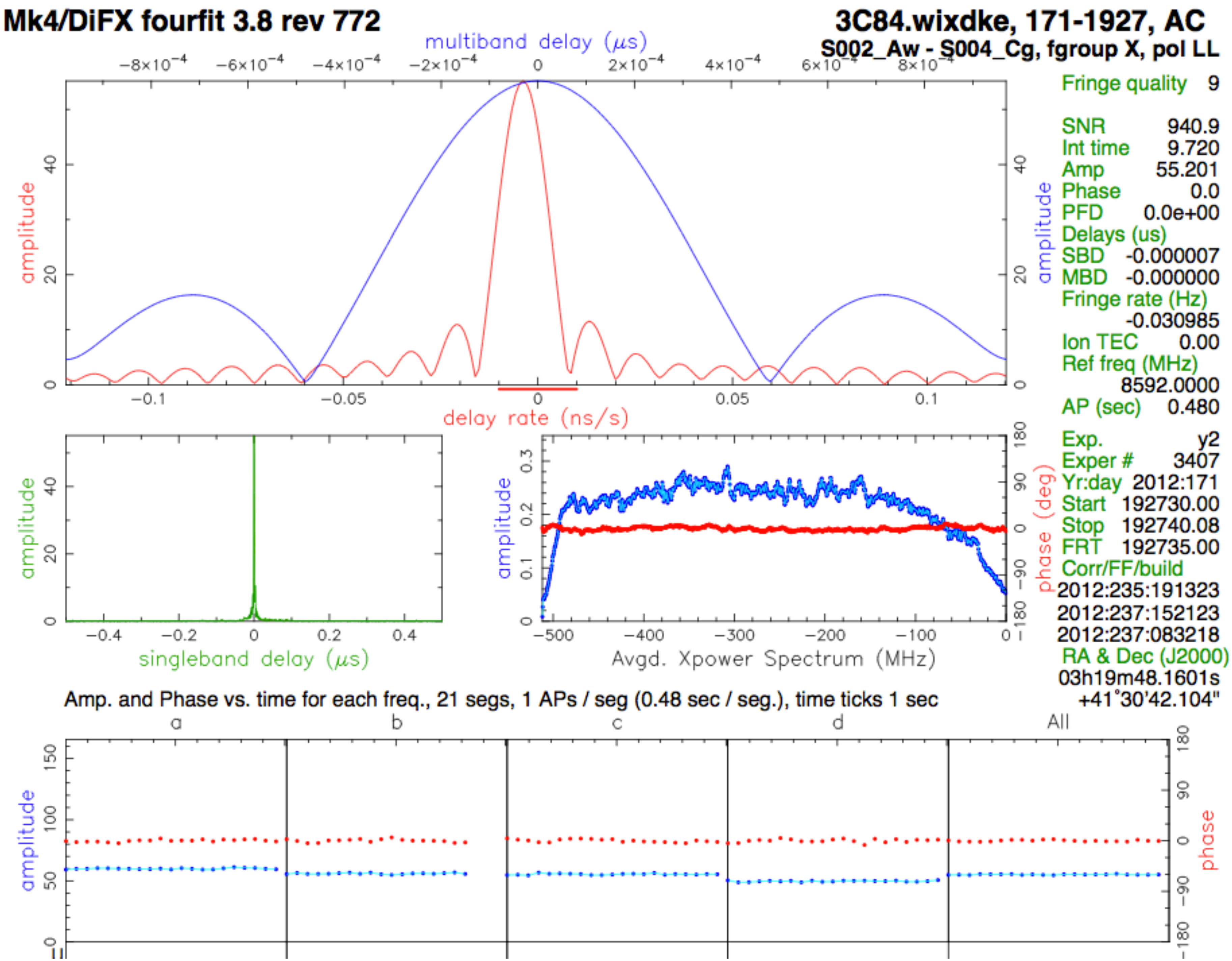
Roger Cappallo
DiFX Developers Meeting
CSIRO, Sydney NSW
2012.9.25

Since last year’s meeting…

- Alan Whitney’s talk on Haystack mtg. site
- prototype software v.0 achieved 16 Gb/s to a RAID array
- author of prototype (David Lapsley) left Haystack
- Roger Cappallo and Chet Ruszczyk writing production version

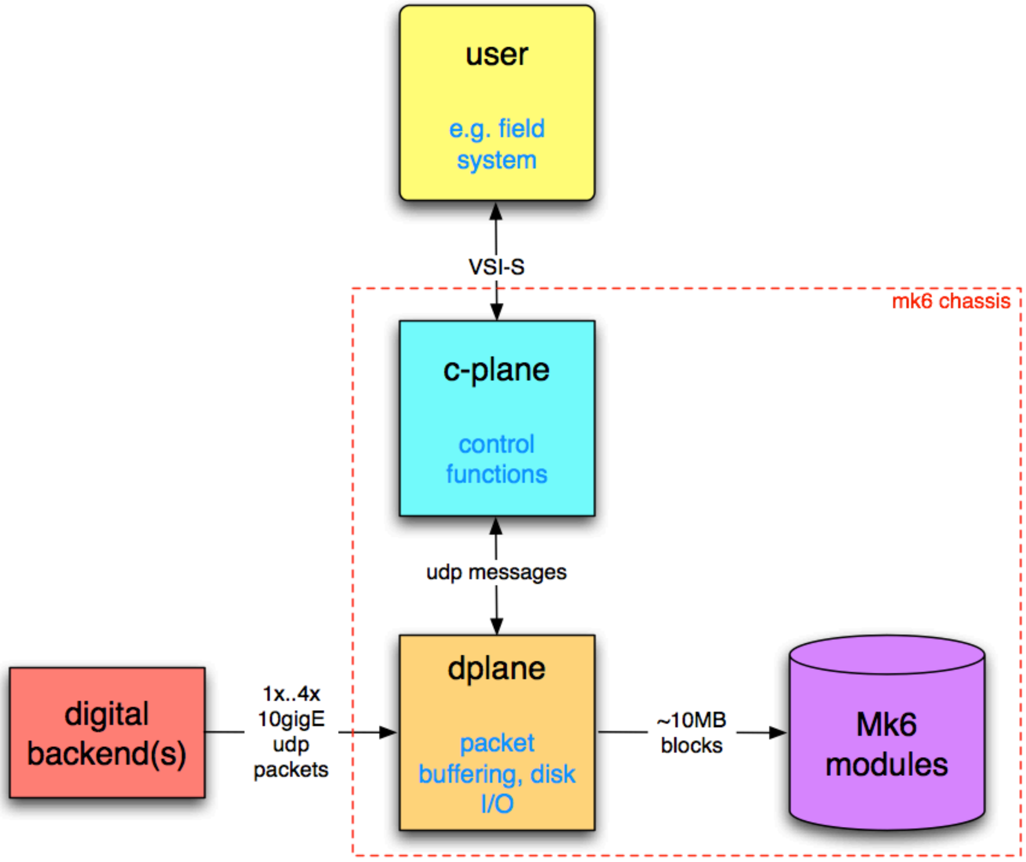
Demonstration Experiment

- June 2012
- Westford – GGAO
- done with prototype software (v.0)
- 16 Gb/s
  - 4 GHz bandwidth on the sky
  - dual polarization with 2 GHz IF’s
  - processed as four 512 MHz channels

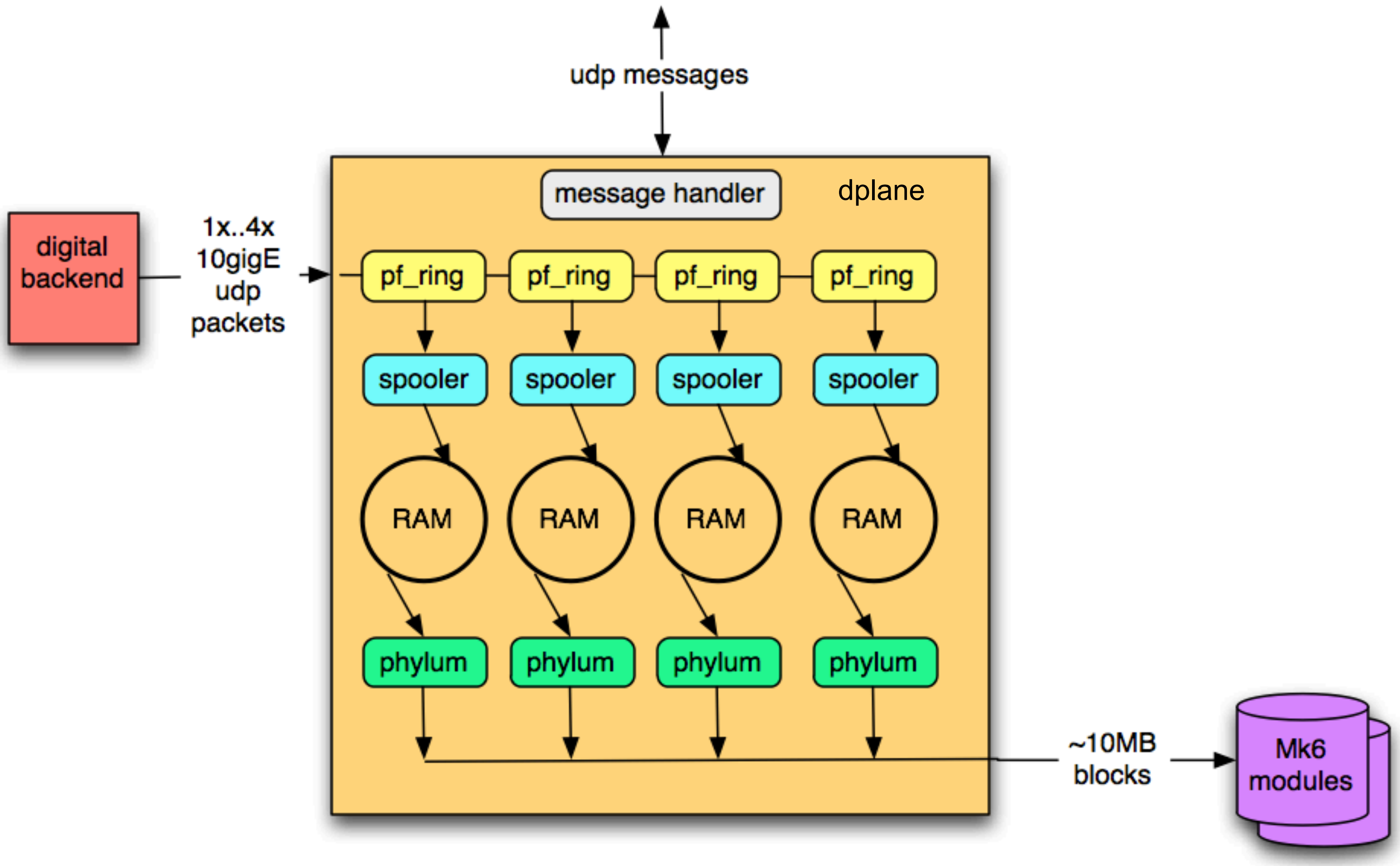


```
Mk4/DiFX fourfit 3.8 rev 772                    3C84.wixdke, 171-1927, AC
                                   S002_Aw - S004_Cg, fgroup X, pol LL

Fringe quality   9

SNR          940.9
Int time     9.720
Amp         55.201
Phase          0.0
PFD        0.0e+00
Delays (us)
SBD      -0.000007
MBD      -0.000000
Fringe rate (Hz)
         -0.030985
Ion TEC       0.00
Ref freq (MHz)
         8592.0000
AP (sec)     0.480

Exp.            y2
Exper #       3407
Yr:day    2012:171
Start    192730.00
Stop     192740.08
FRT      192735.00
Corr/FF/build
2012:235:191323
2012:237:152123
2012:237:083218
RA & Dec (J2000)
03h19m48.1601s
+41°30'42.104"

Amp. and Phase vs. time for each freq., 21 segs, 1 APs / seg (0.48 sec / seg.), time ticks 1 sec
```

mark6 block diagram



dplane block diagram



dplane - Technical Highlights

- **pf_ring** used for high-speed packet buffering
- **smp affinity** of IRQ's, and **thread binding** to cores based on #of available cores
- most of physical RAM grabbed for large ring buffers, one per stream, and locked in
- non-blocking I/O using select() calls
- 10 MB blocks scattered to files resident on different disks
  - prepended block# for ease of reassembly
  - balances disk usage as much as possible
- *gather* uses asynchronous I/O to read n disks into n ring buffers, and write in order to single file

Miscellaneous Considerations

- capture to ring buffers is kept separate from file writing, to facilitate eVLBI, etc.
- FIFO design decouples writing from capturing (e.g. keep writing during slew)
- all mk6 software is open source for the community

Timeline

- Current Status
  - functional control plane and v.2 data plane software, still being integrated

- Timeline
  - Dec 2011: v.0 prototype achieved 16 Gb/s to 4x8 disk RAID arrays
  - July 2012: v.1 operational dataplane code using RAID array
  - Sept 2012: integration of v.1 control and data plane codes
  - Sept 2012: v.2 dataplane code with scattered filesystem

now

  - Oct 2012: complete integration of control and v.2 dplane
  - Oct-Nov 2012: performance testing, assessment, & tuning
  - Jan 2013: first operational broadband use (8 Gb/s on 1 module)

Playback Strategies

v.0 prototype

v.1 operational RAID-based code

v.2 with single output file per stream:

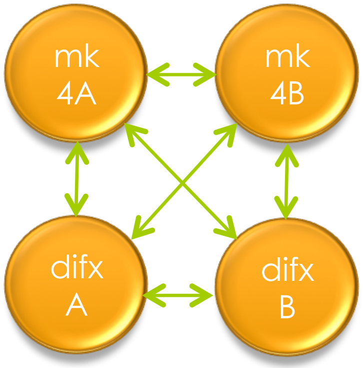
- write (ordinary) Linux files on RAID arrays, can use normal file-based correlation

v.2 with multiple files data needs to be reconstituted:

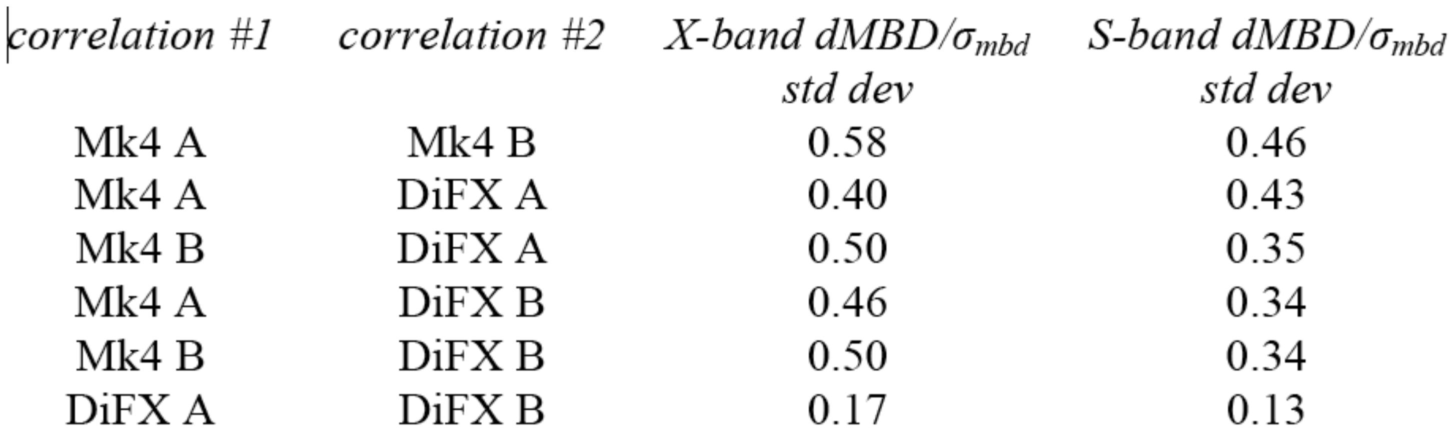
- *gather* program does so very efficiently, but it requires an extra step
- might write a FUSE/mk6 interface
- will likely implement native mk6 datastream

mk4 vs. difx intercomparison

- by Brian Corey and Mike Titus
- memo posted on difx web pages:
  (http://cira.ivec.org/dokuwiki/doku.php/difx/difx2mark4)
- multiband delays
  - X-band
  - S-band
  - 16 scans
- correlation:
  - mk4 hardware correlator
  - difx software correlator
  - each with 2 different models (clock offset & rate)
- rms of total multiband delay differences



rms of Δ multiband delay



correlation #1	*correlation #2*	*X-band dMBD/σ_{mbd}*	*S-band dMBD/σ_{mbd}*
		std dev	*std dev*
Mk4 A	Mk4 B	0.58	0.46
Mk4 A	DiFX A	0.40	0.43
Mk4 B	DiFX A	0.50	0.35
Mk4 A	DiFX B	0.46	0.34
Mk4 B	DiFX B	0.50	0.34
DiFX A	DiFX B	0.17	0.13