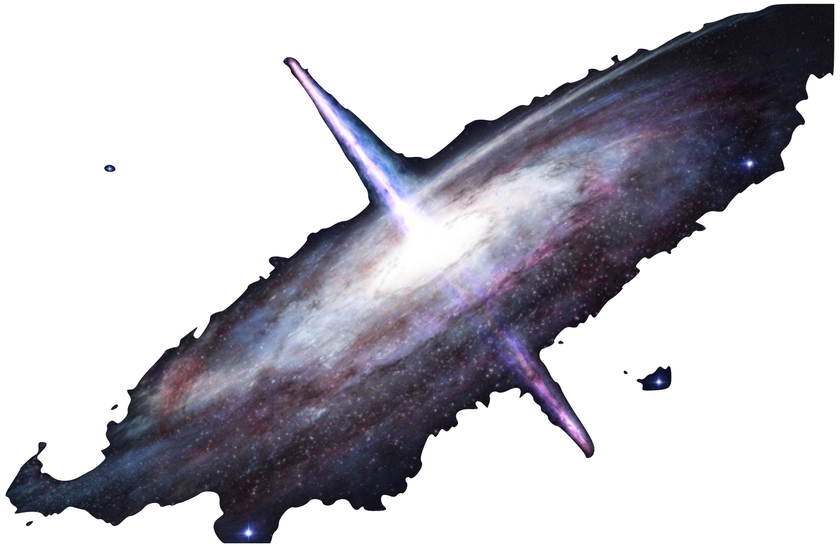
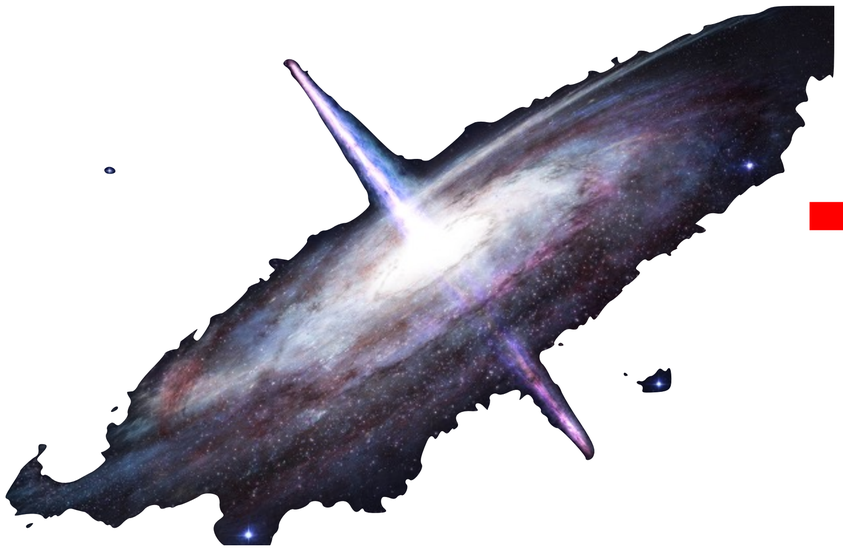
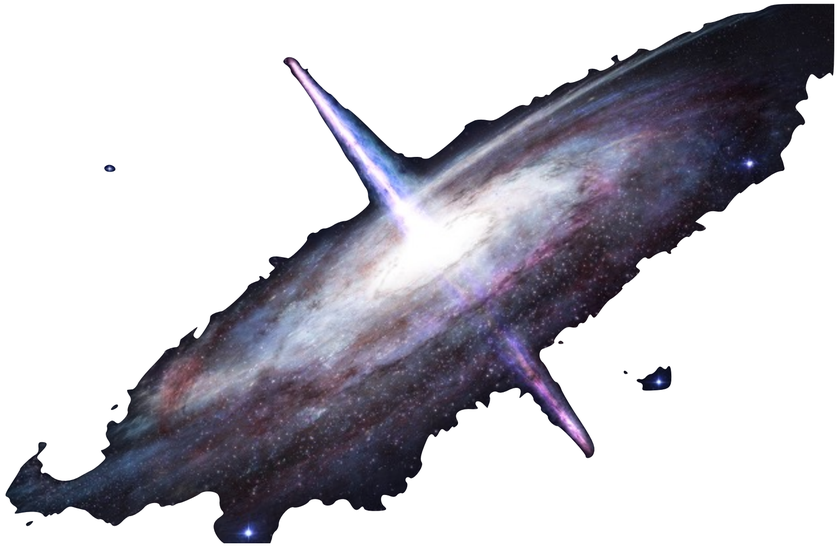
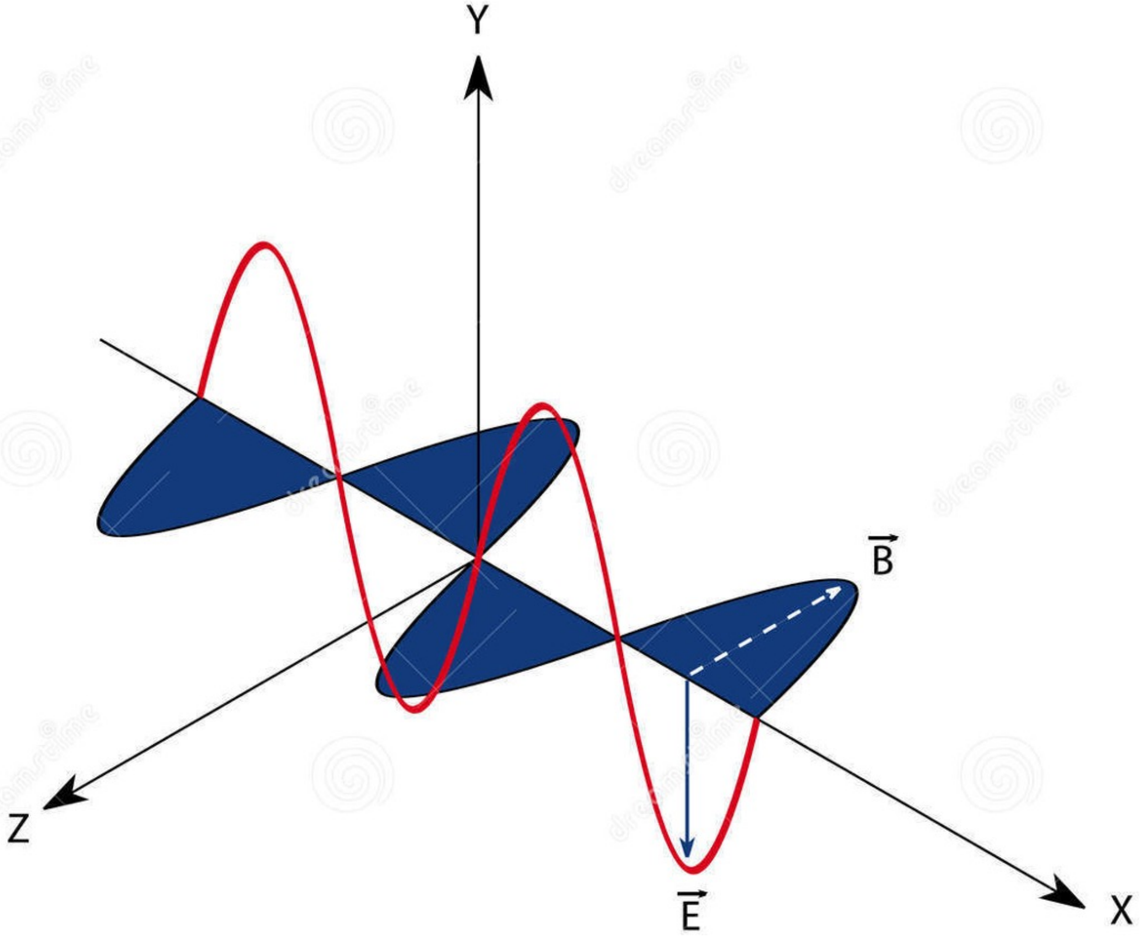
$$\sigma_T = \frac{T_{sys}}{\sqrt{\Delta_\nu \cdot \Delta t}}$$

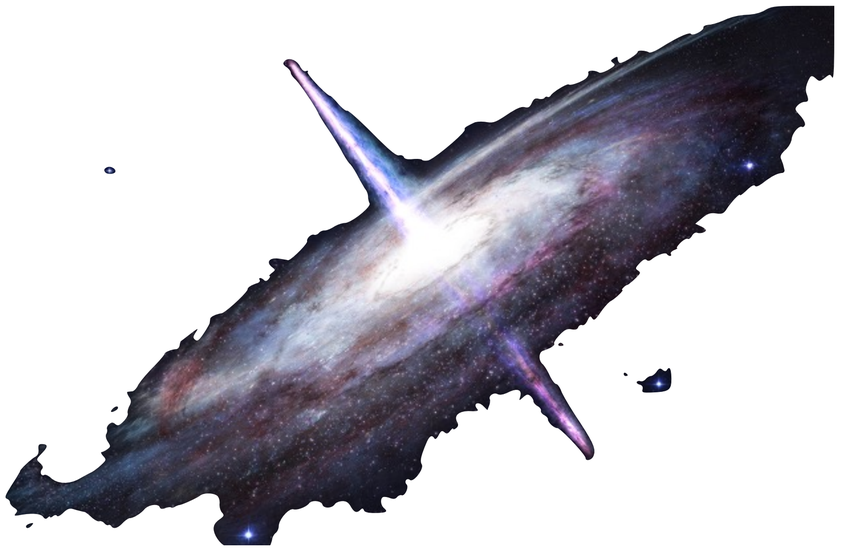


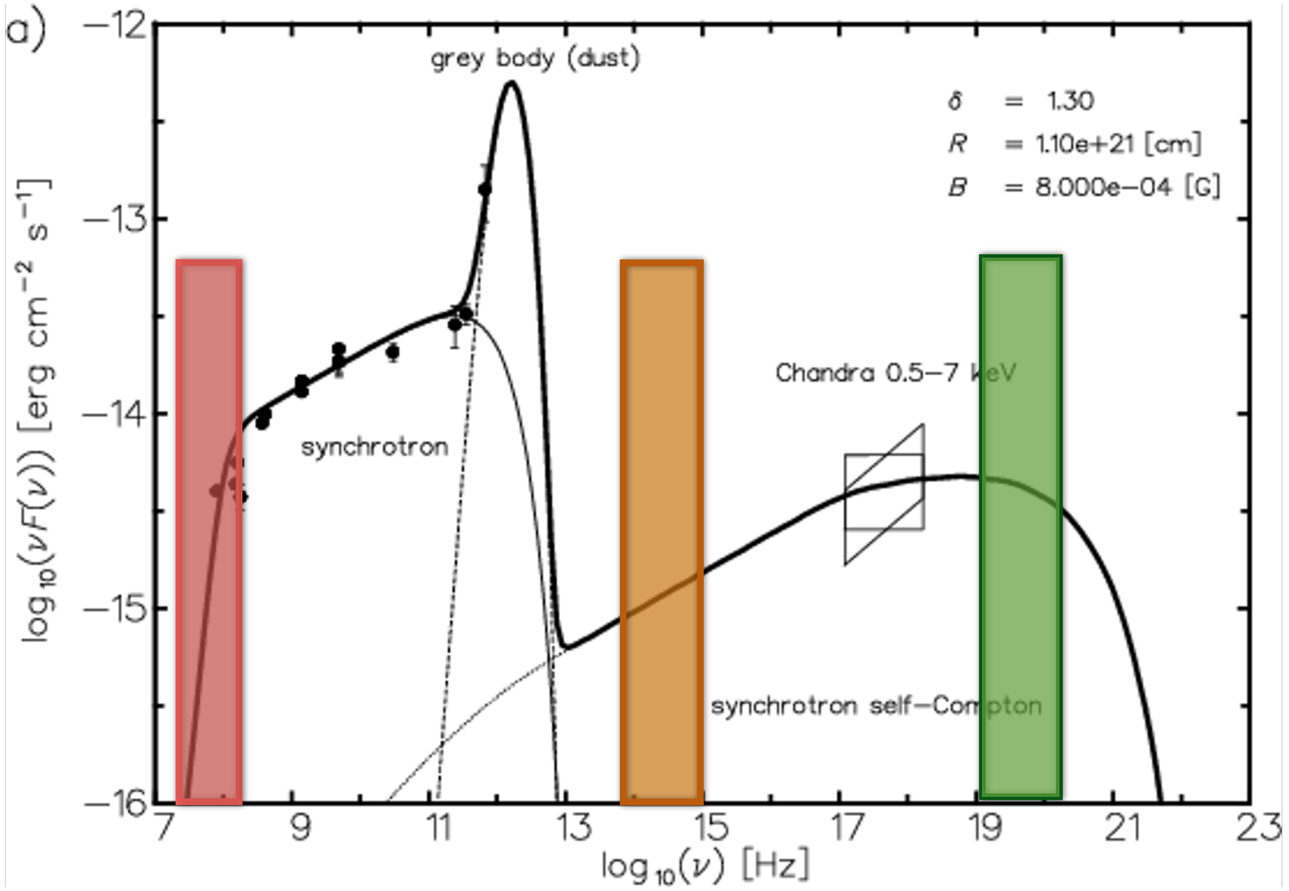



WD740




Y
B
Z
E
X

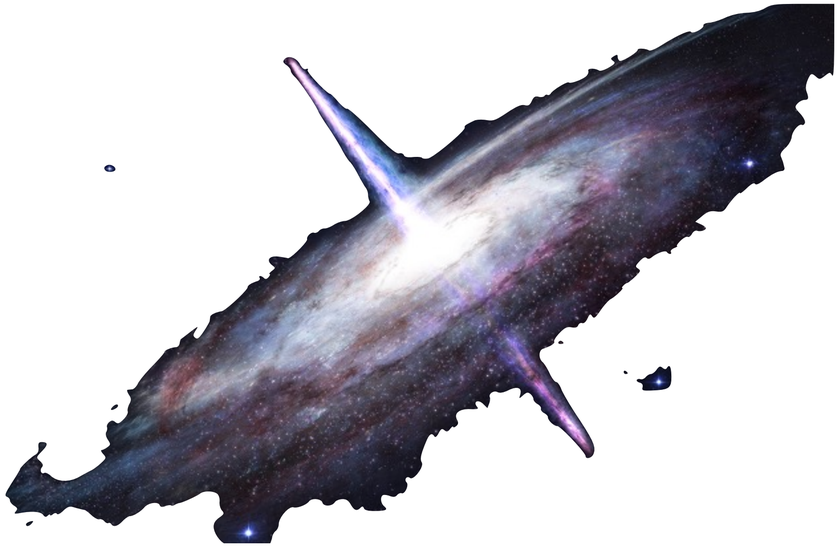


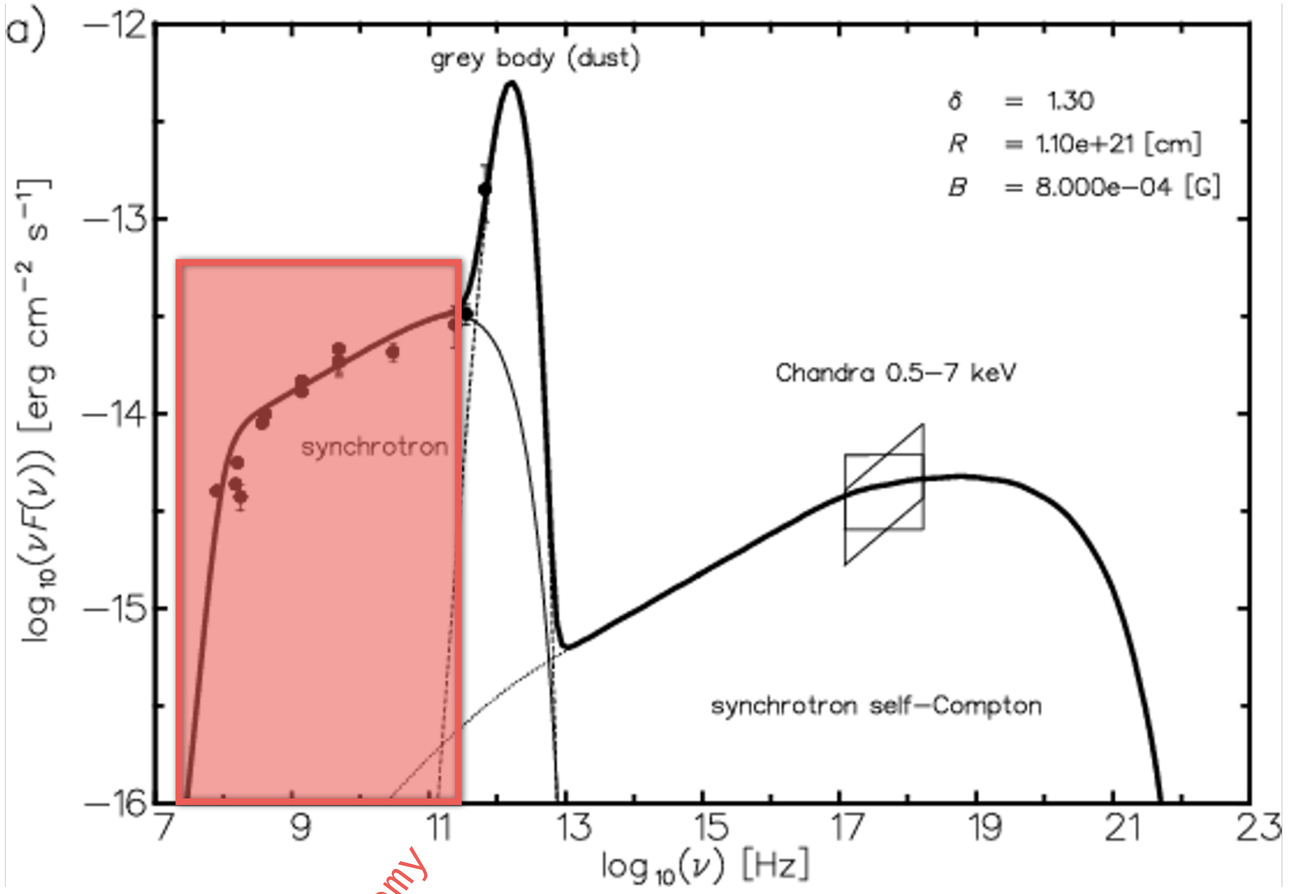


FM radio television

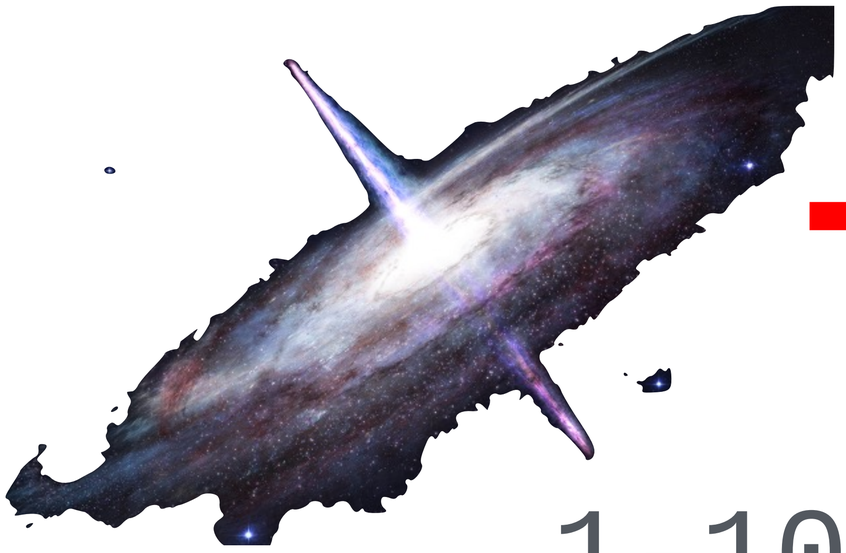
optical

gamma



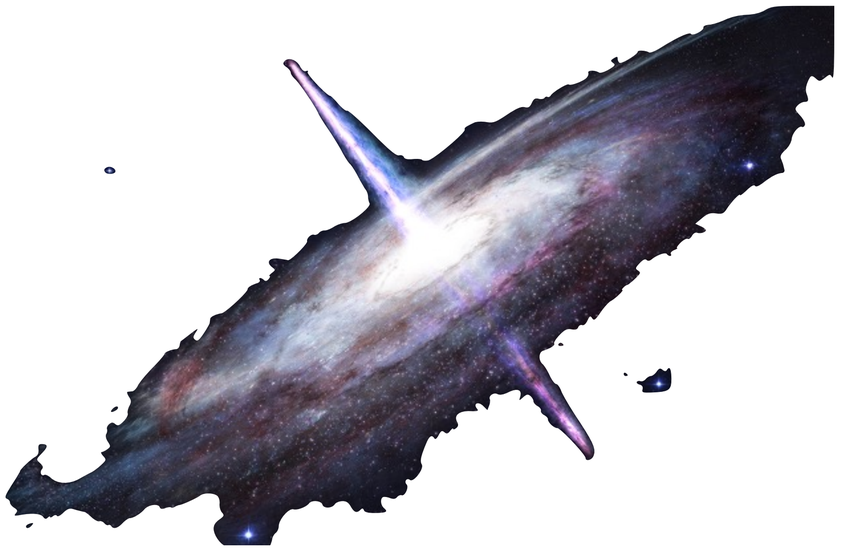


radio astronomy



1-100 GHz RF signals

Undesirable to sample/digitize/process as one chunk!



?

analog

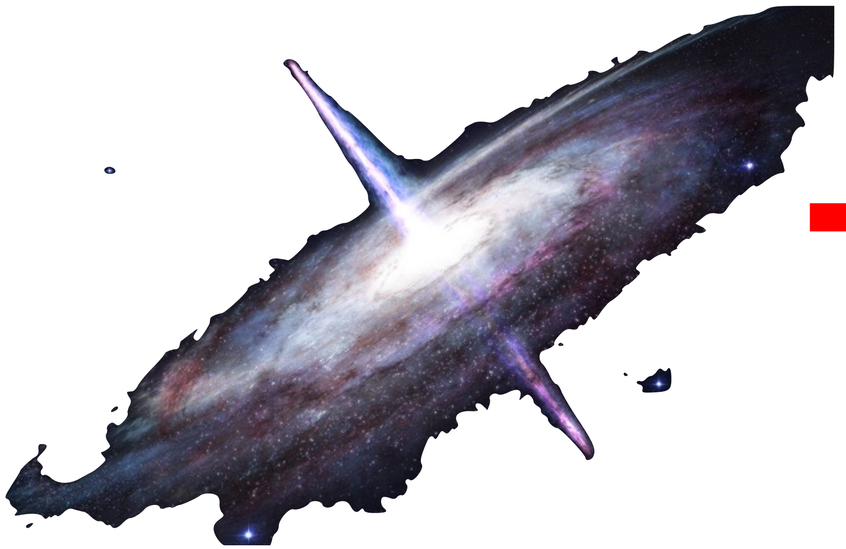
digital

VLBI Data:

acquisition, formats, transport and tools

JIVE
Joint Institute for VLBI
ERIC

Marjolein Verkouter



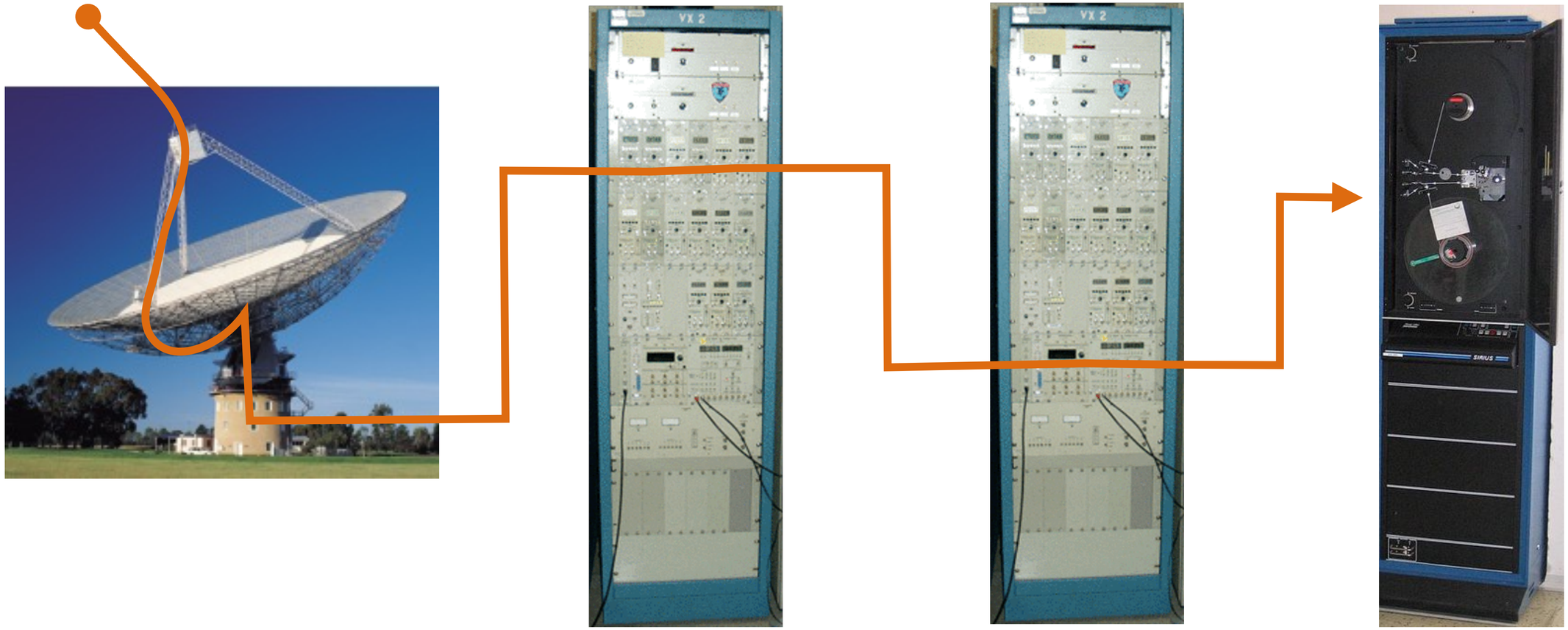
analog

digital


VX 2

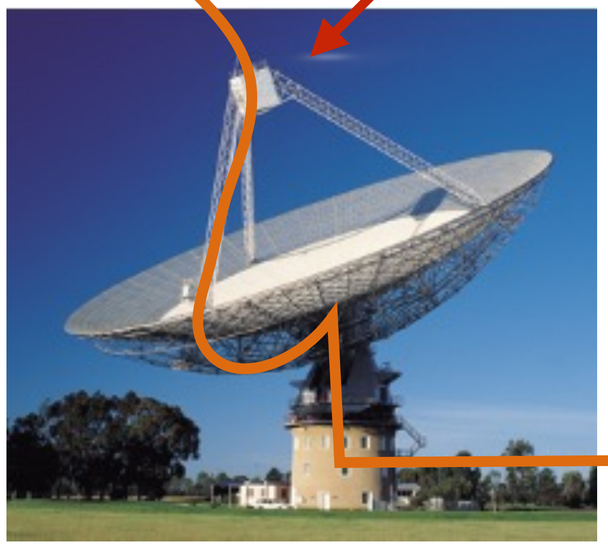

VX 2


SIRIUS


VX 2
VX 2
SIRIUS

sky freq
1-100 GHz

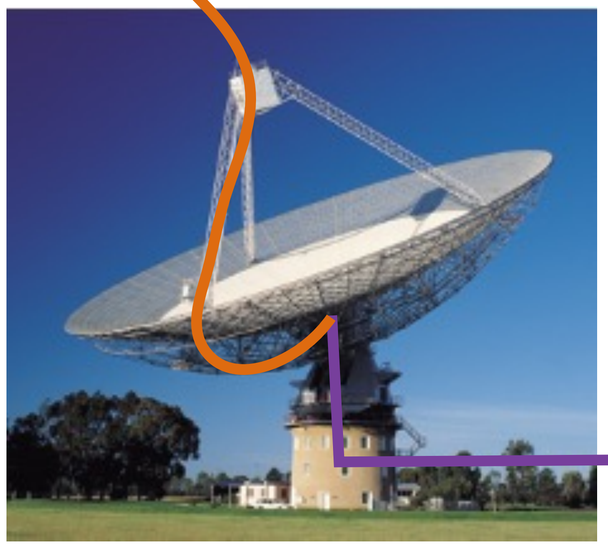
telescope front end



mix sky with LO

sky freq
1-100 GHz

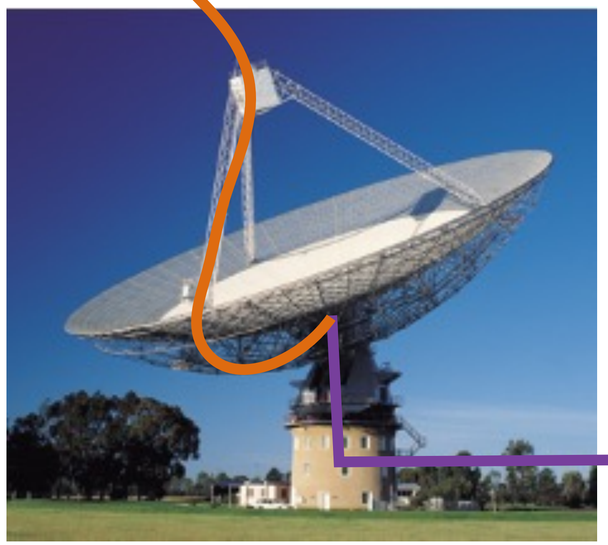
IF
0-1 GHz



mix sky with LO

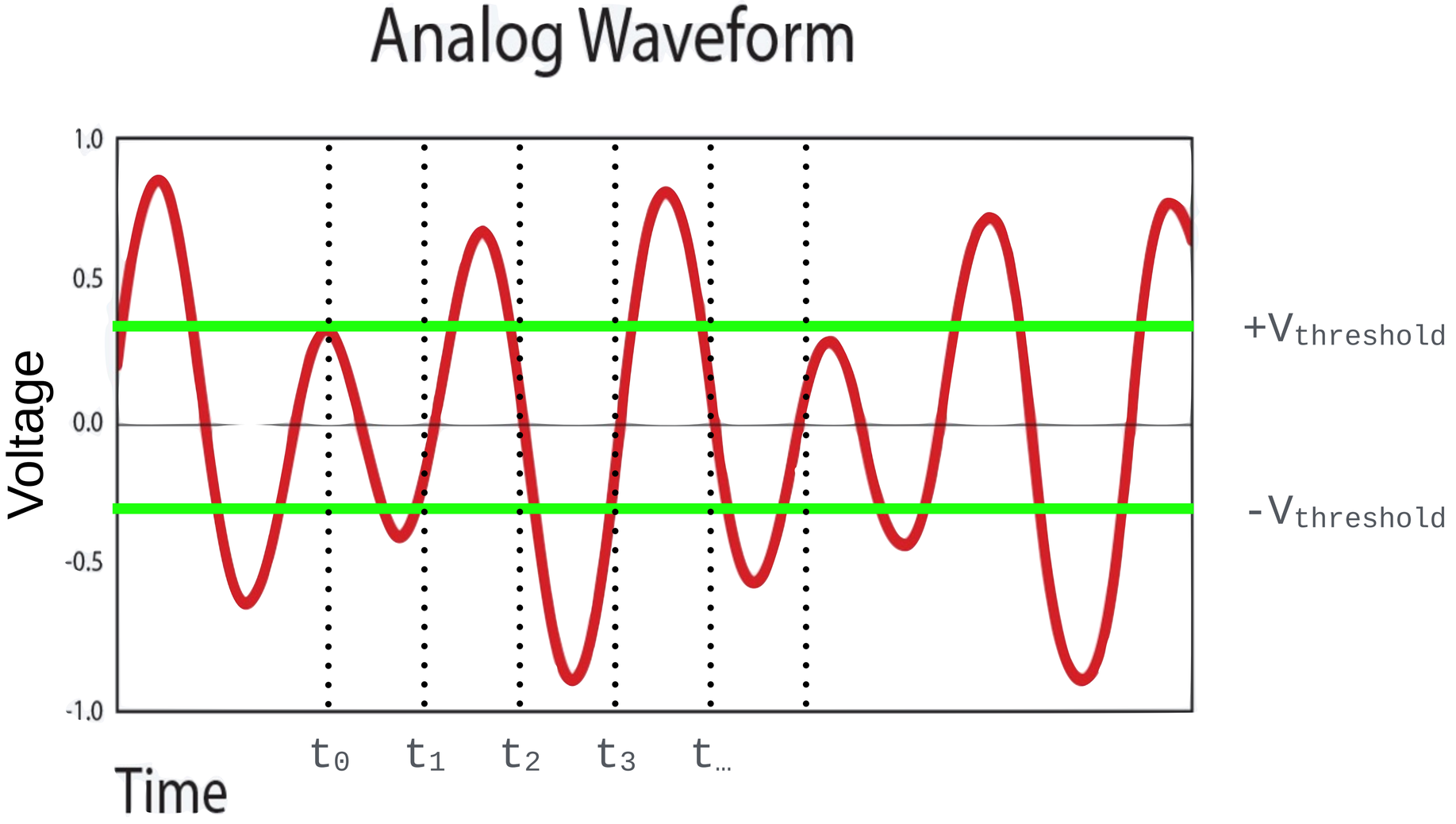
sky freq
1-100 GHz

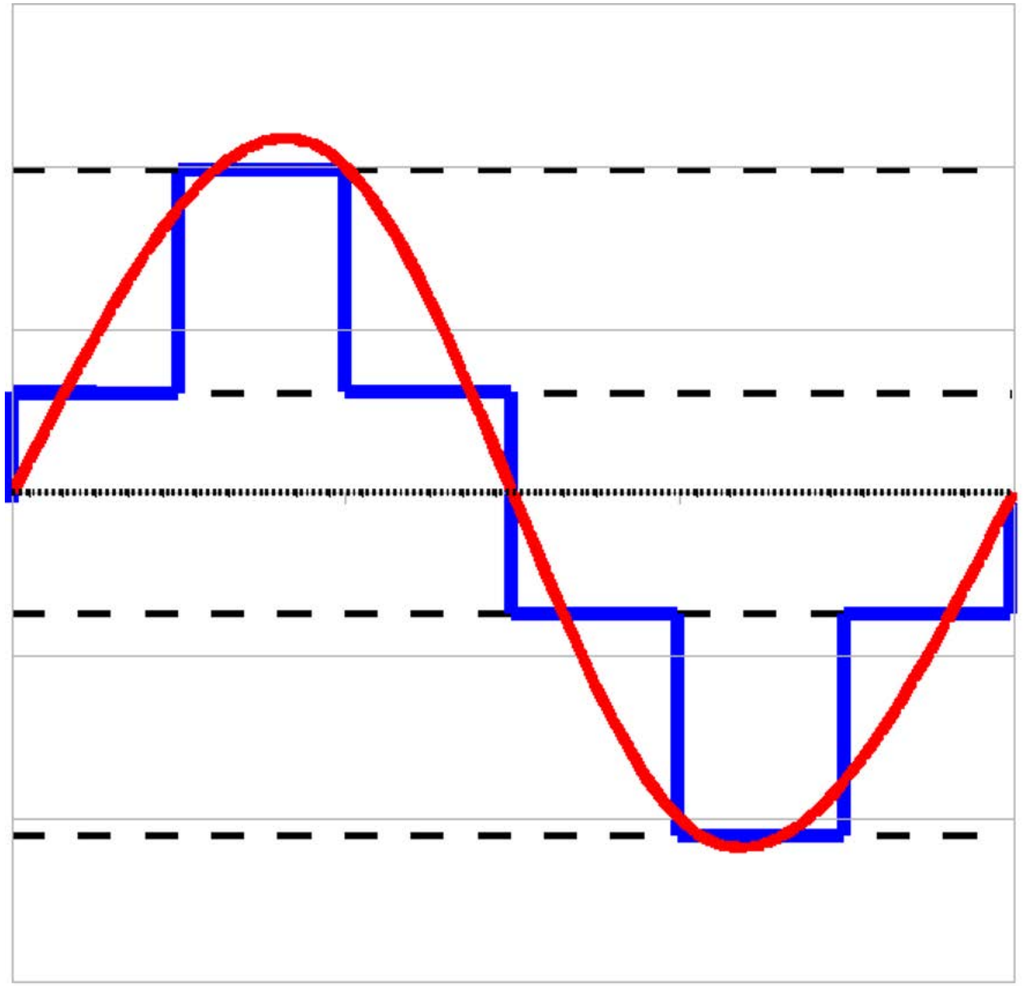
IF
0-1 GHz



mix sky with LO

BBC, A-to-D conversion


Analog Waveform
1.0
0.5
0.0
-0.5
-1.0
Voltage
Time
t0
t1
t2
t3
t…
+Vthreshold
-Vthreshold

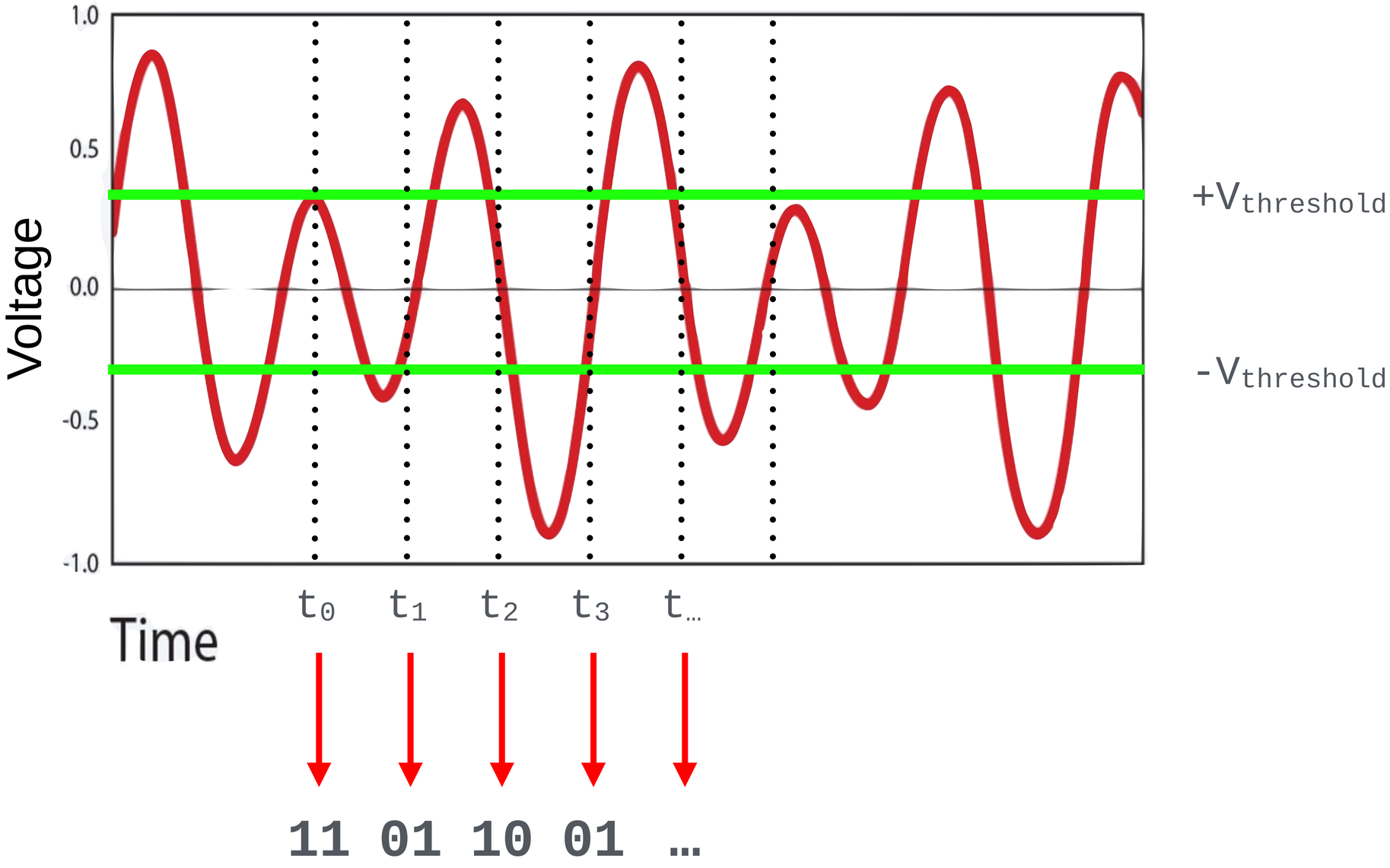


11
10
01
00

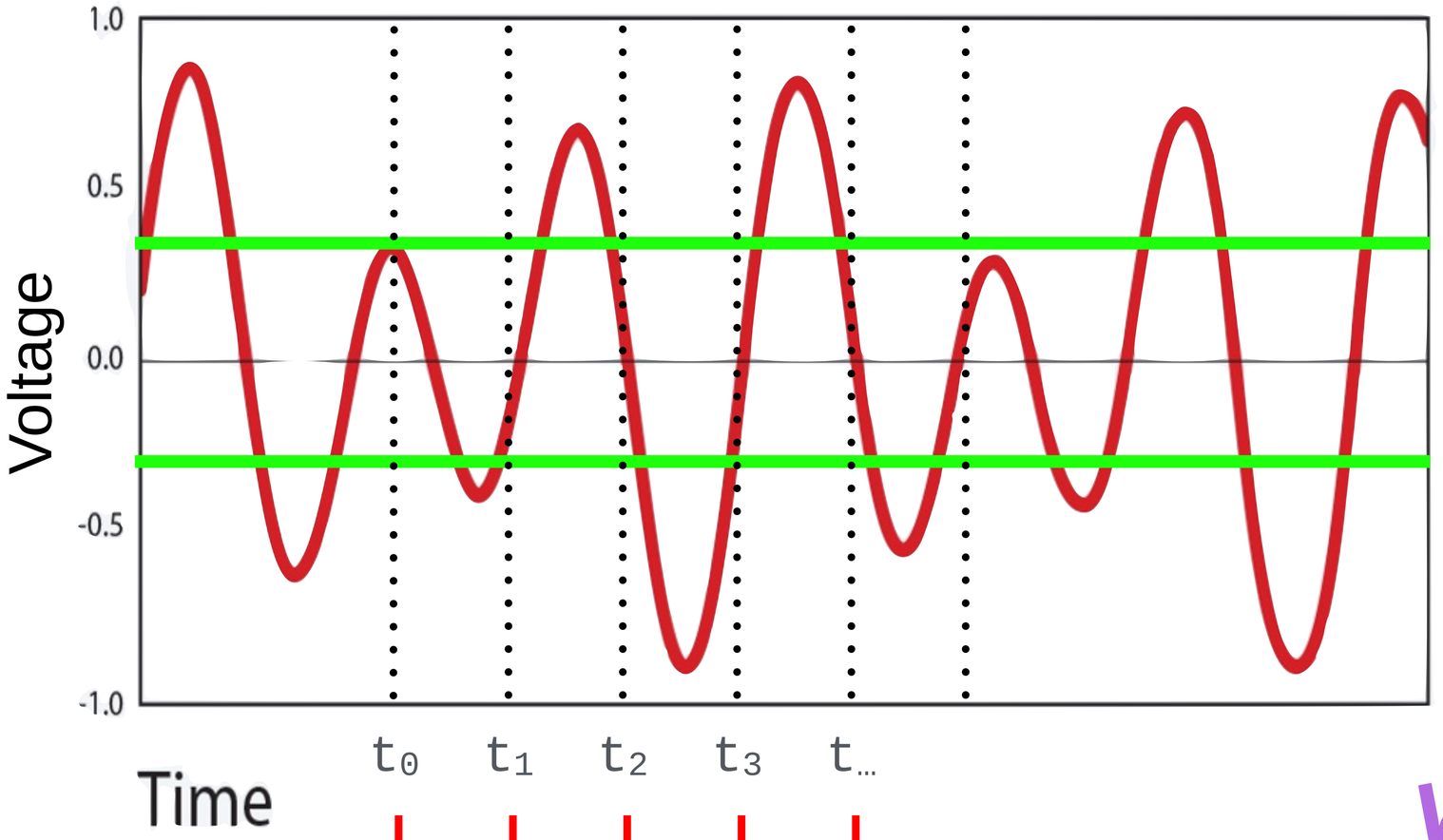
2-bit
binary
encoded
value

analog signal
discretized signal

Analog Waveform



Analog Waveform



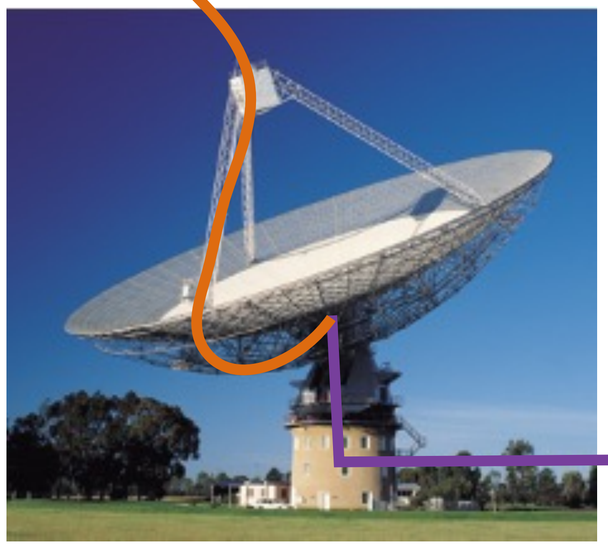
11 01 10 01 …

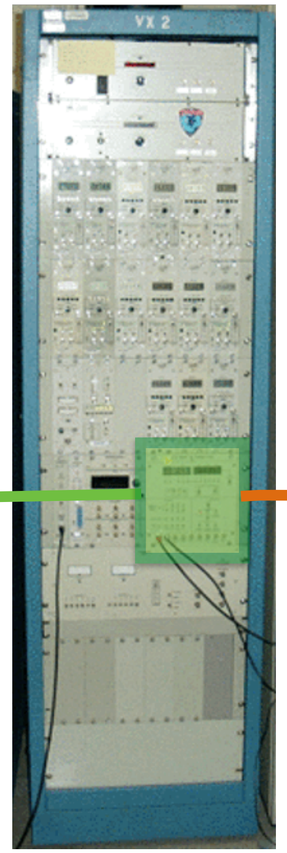
bitstream

sky freq
1-100 GHz

IF
0-1 GHz

bitstreams
n·10⁶ Bps





mix sky with LO

BBC, A-to-D conversion

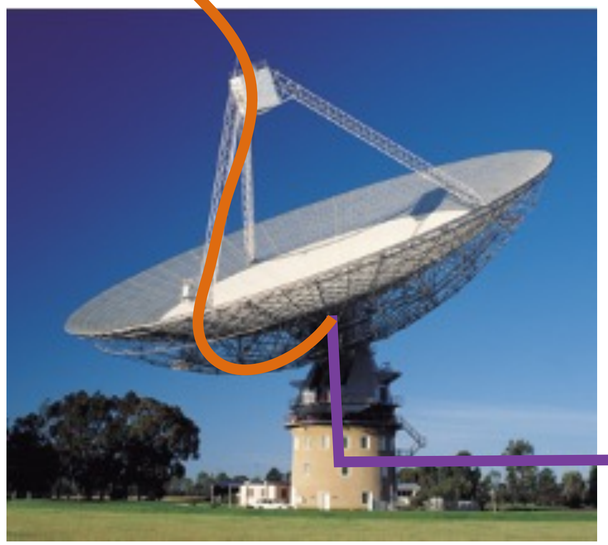
formatting

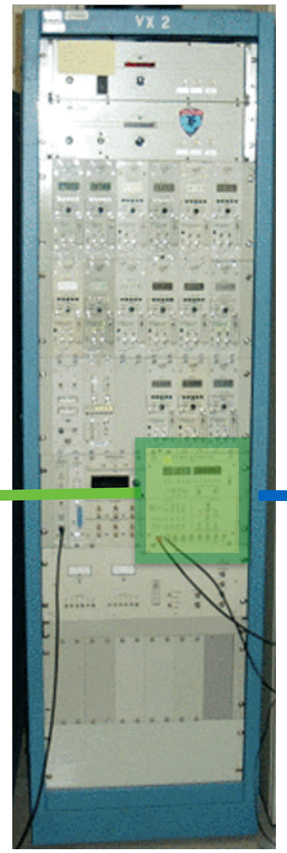
sky freq
1-100 GHz

IF
0-1 GHz

bitstreams
n·10⁶ Bps

VLBI format
dataframes





mix sky with LO

BBC, A-to-D conversion

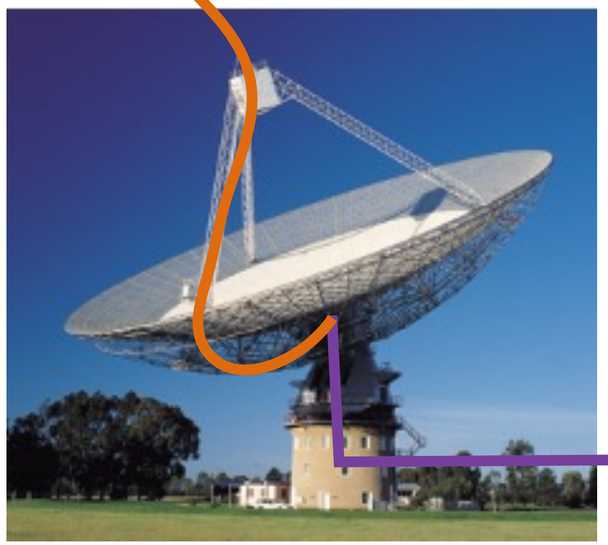
formatting

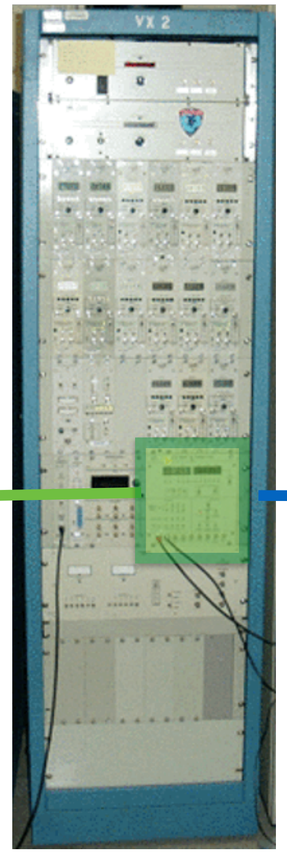
sky freq
1-100 GHz

IF
0-1 GHz

bitstreams
n·10⁶ Bps

VLBI format
dataframes





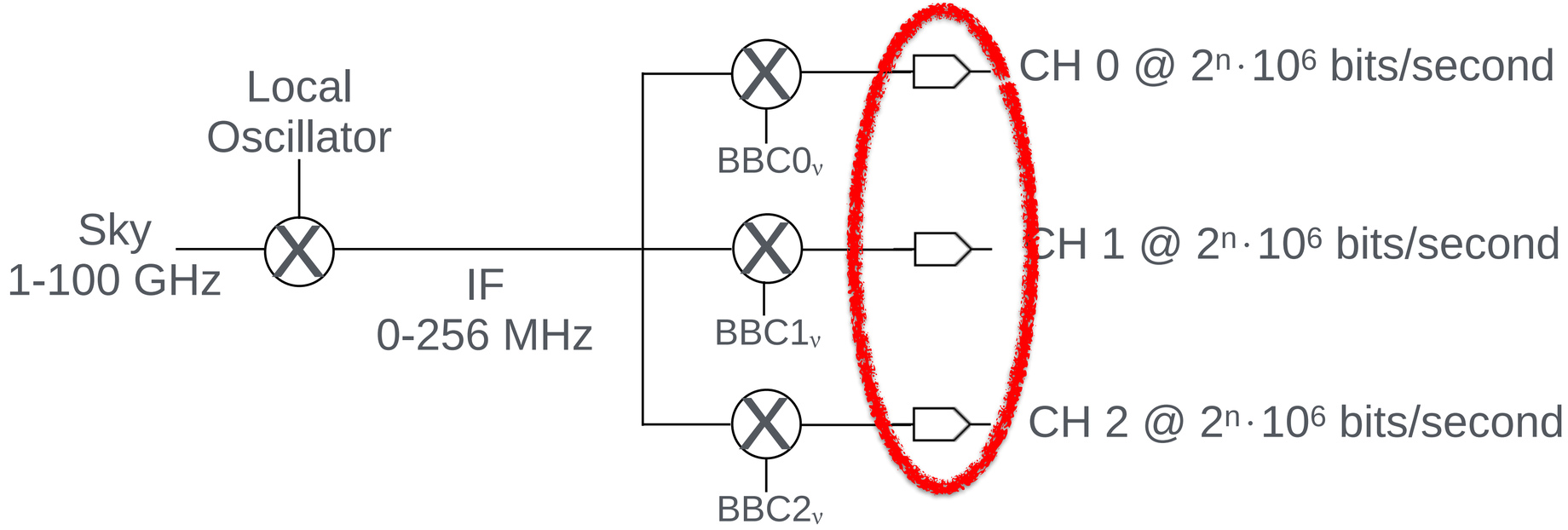
mix sky with LO

BBC, A-to-D conversion

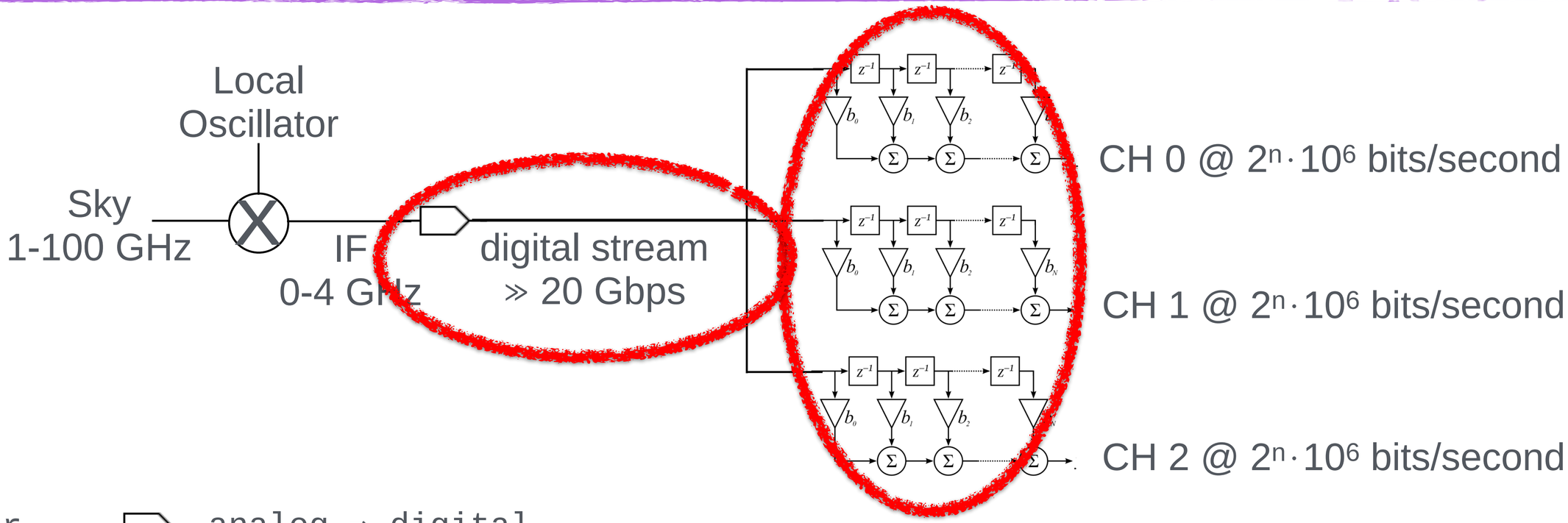
formatting

recording

Old



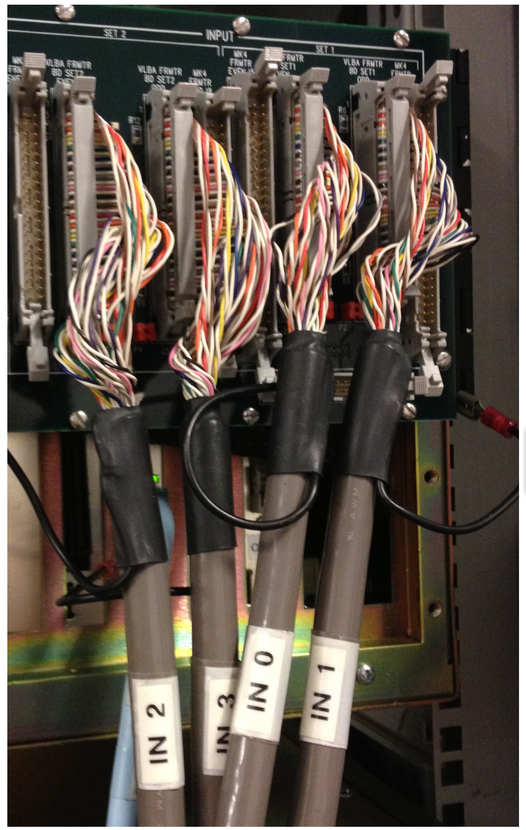
New



mixer

analog ⇒ digital

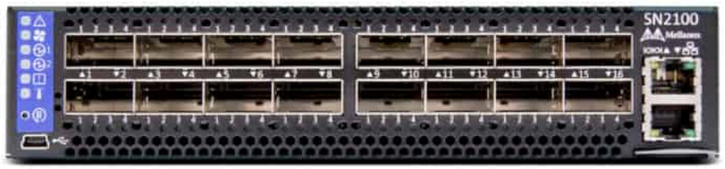
Analog Backend



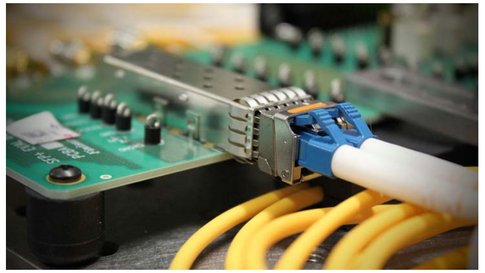
64 x RS422
at (max) 10 Mbps / link

Recorder

n x CX4
at 10 Gbps / link



Digital Backend



Recorder

Recorder

n x SFP+/fibre
at 10 Gbps / link

Getting as many bits on disk as possible

$$\sigma_T = \frac{T_{sys}}{\sqrt{\Delta_\nu \cdot \Delta t}}$$

more = better

$$Sensitivity = \frac{1}{\sigma_T}$$

$$f_{sample} = 2 \cdot \Delta_{\nu}$$

https://en.wikipedia.org/wiki/Nyquist–Shannon_sampling_theorem

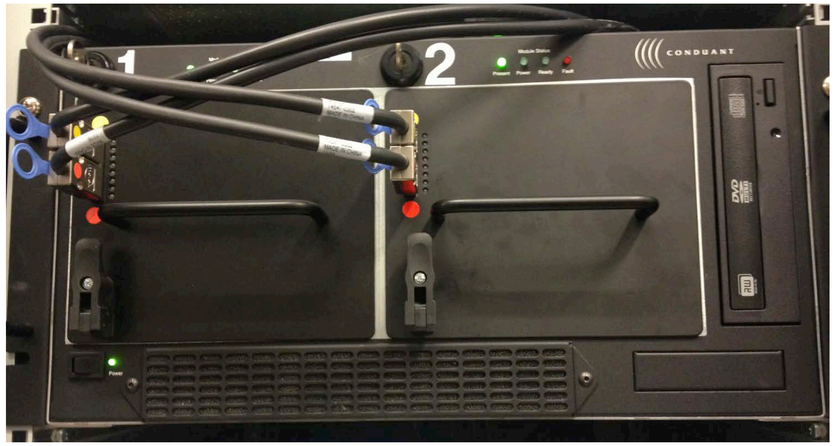
Contemporary VLBI recorders

ethernet (UDP/IP) packets

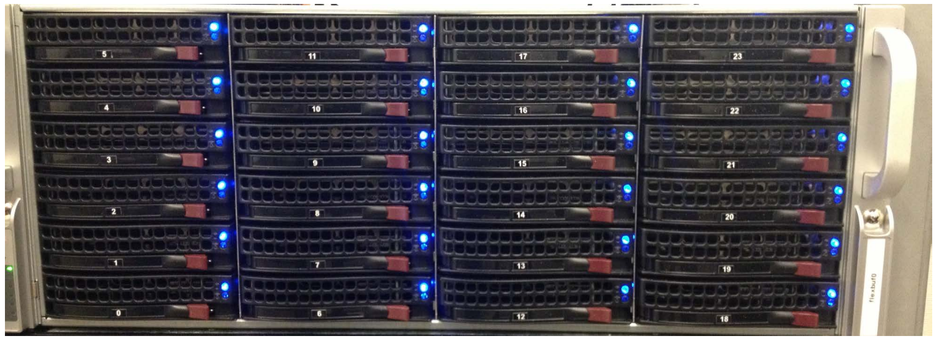
- $2^n \cdot 10^6$ Mbps
- real sampled
- VDIF(*)

(*) VLBI Data Interchange Format - https://vlbi.org/vlbi-standards/vdif/

Contemporary VLBI recorders



Mark6

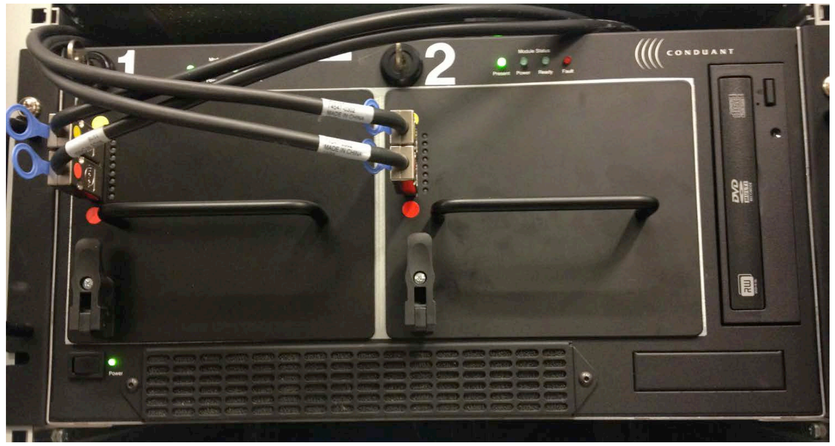


FlexBuff

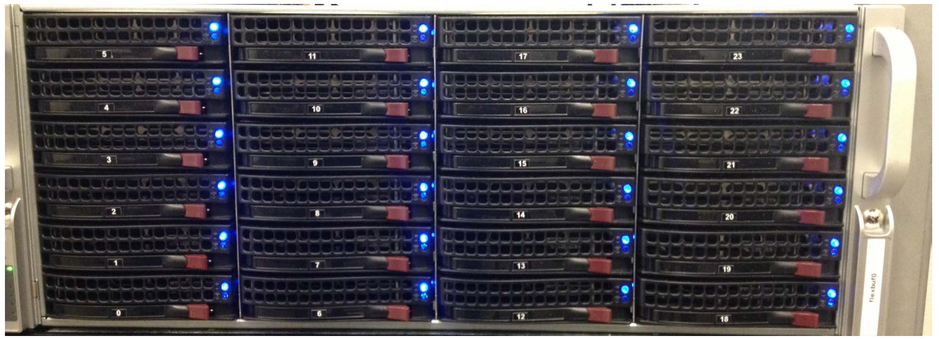
OCTADISK

...?

Contemporary VLBI recorders



Mark6



FlexBuff

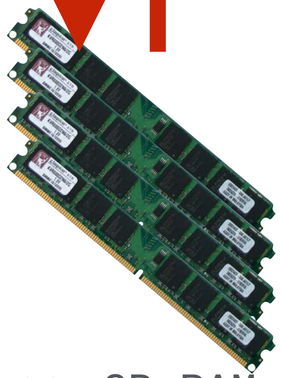
OCTADISK

...?

Contemporary ethernet recorders

ethernet cards

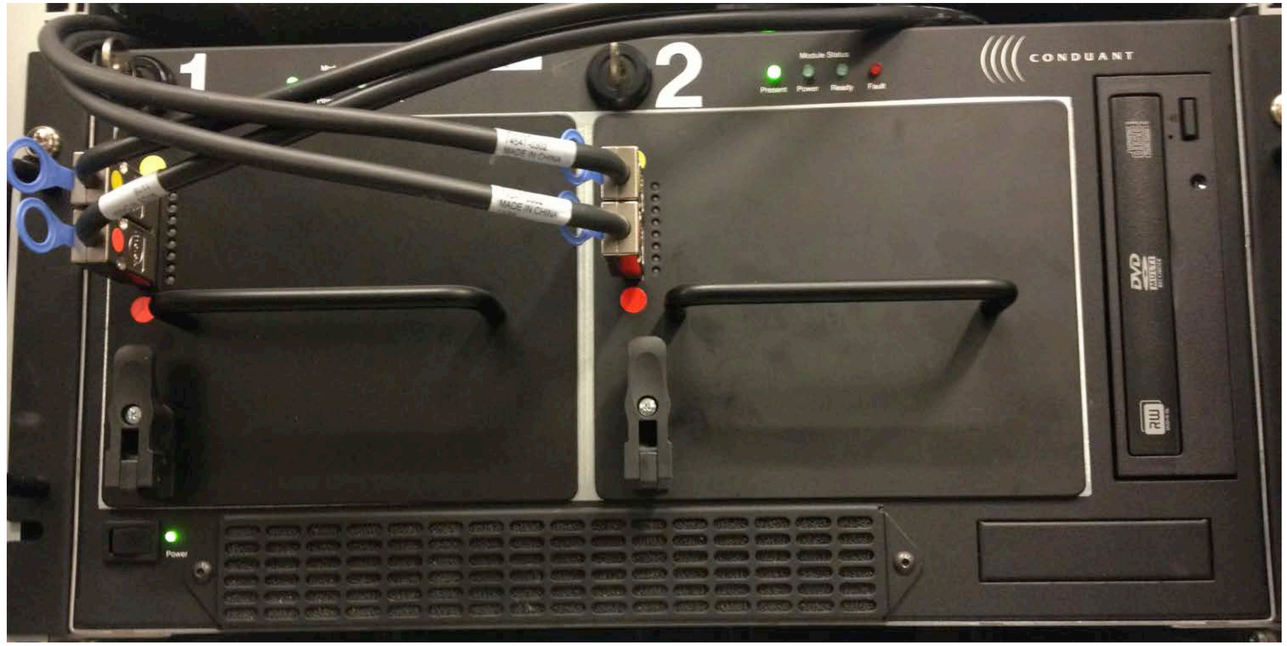
CPUs



>> GB RAM

many HardDisks

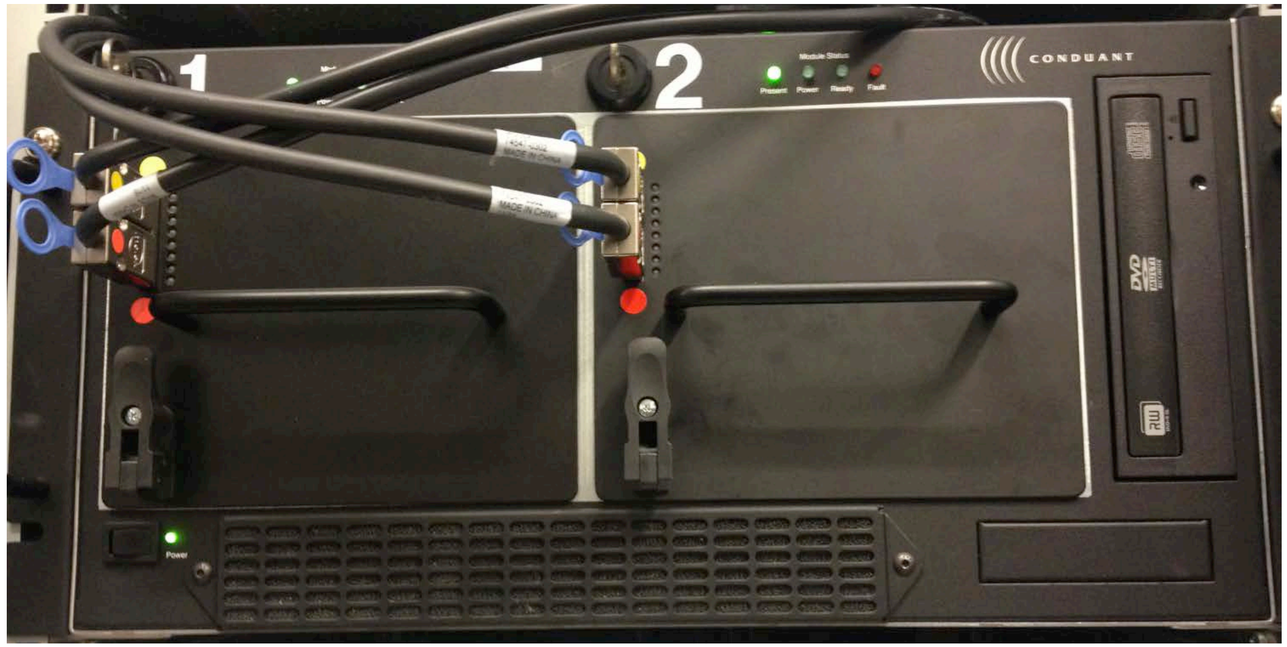
Contemporary ethernet recorders



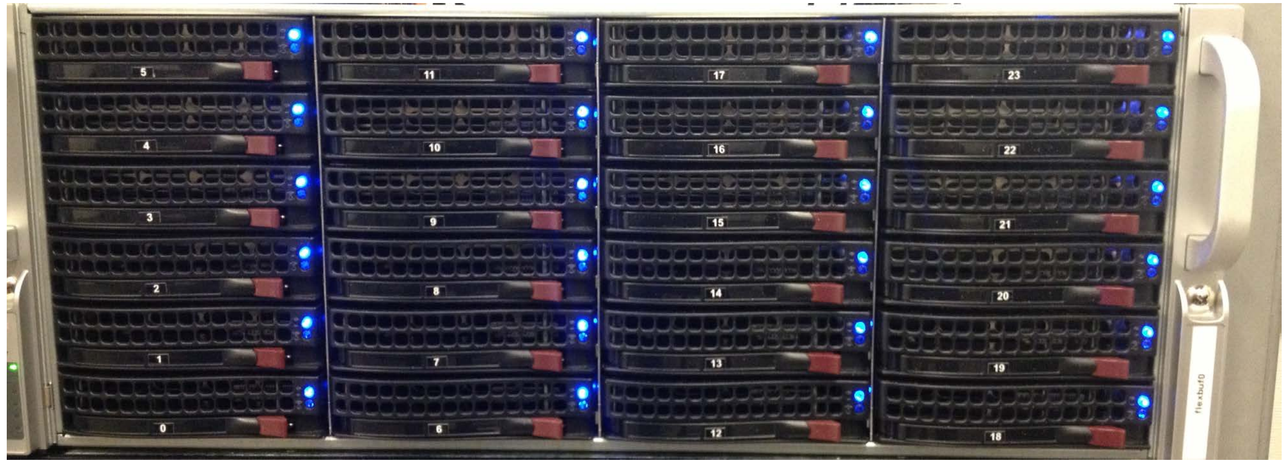
Mark6 (MIT Haystack/Conduant)

- proprietary hardware
- only one supplier (Conduant Corp.)
- ≦ 8 Gpbs
- 30 k€ (inc. 32 x 10 TB HDD)

Contemporary ethernet recorders



Mark6 (MIT Haystack/Conduant)
- proprietary hardware
- only one supplier (Conduant Corp.)
- ≦ 8 Gpbs (≦ 16 Gbps with expander)
- 30 k€ (32 x 10 TB HDD + expander)



FlexBuff (Metsähovi / JIVE)
- fully customizable, fully COTS
- *n* Gpbs
- ~20 k€ (inc. 36 HDD)

Concept: A. Mujunen, Metsähovi
Productionalized: JIVE

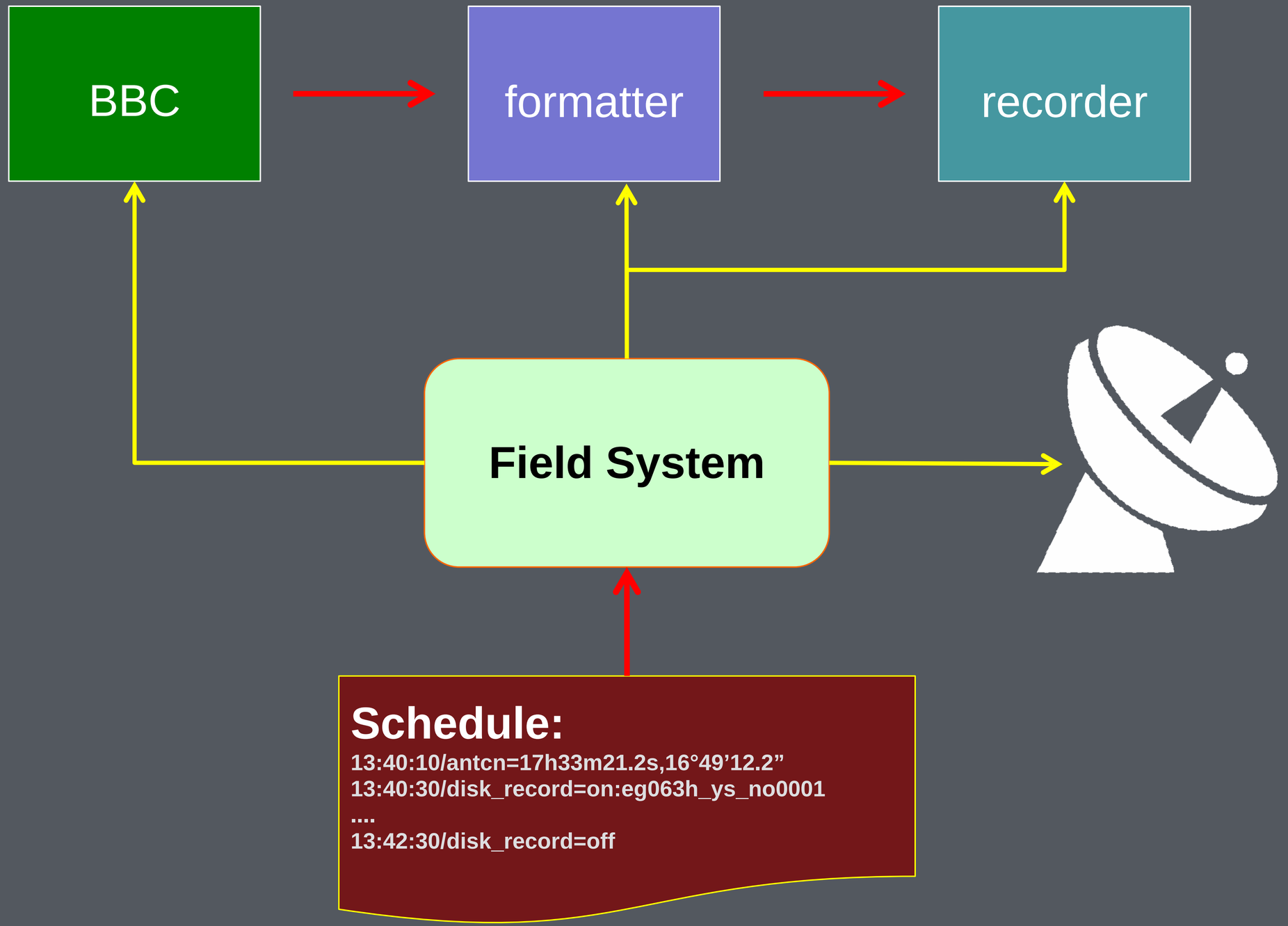
Contemporary ethernet recorders

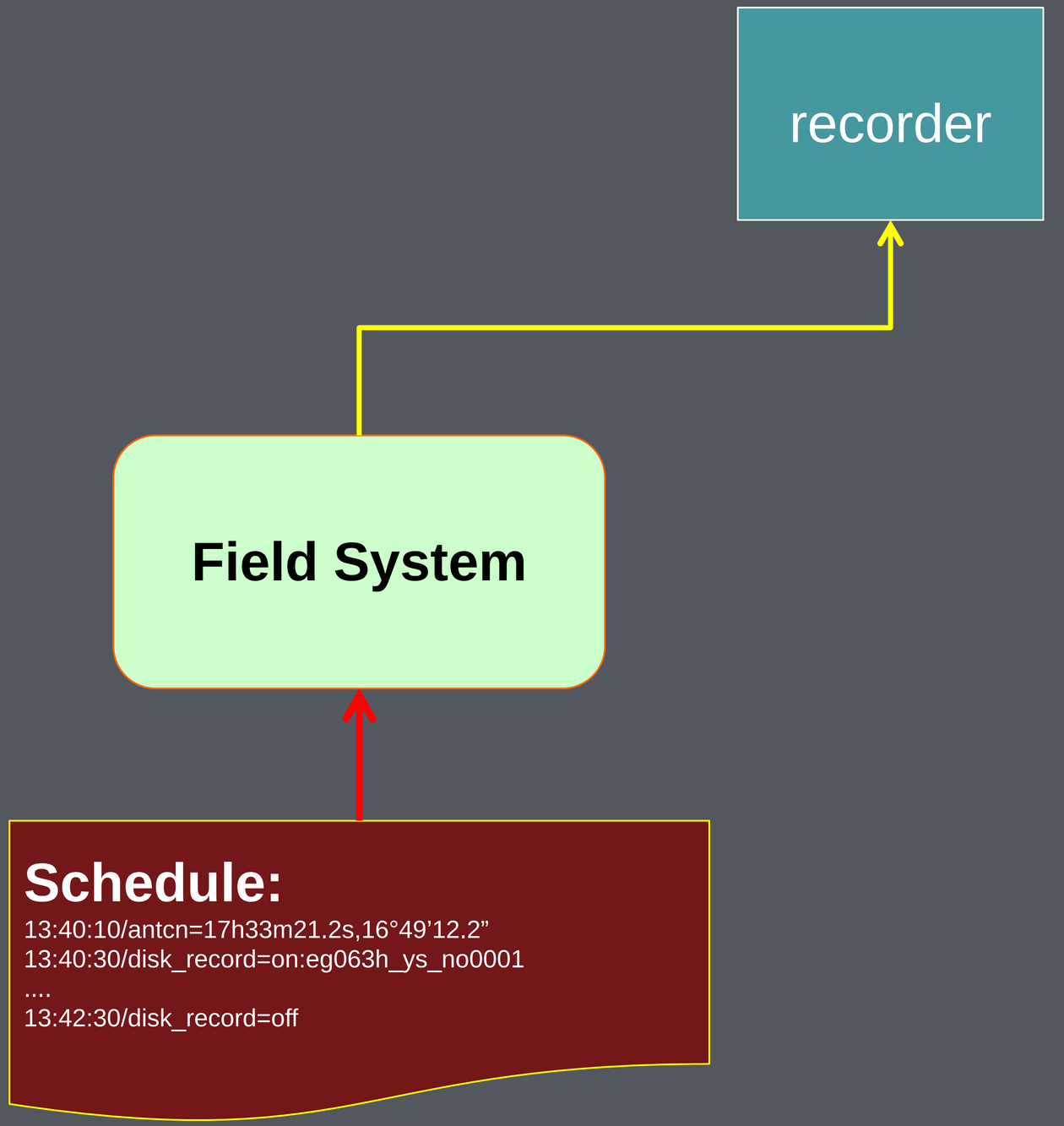
The only tangible difference between the systems. The rest is semantics/software.

Mark6 removable disk packs

FlexBuff fixed disks


And now for something
completely different.
2:39 / 1:28:19


BBC
formatter
recorder
Field System
Schedule:
13:40:10/antcn=17h33m21.2s,16°49’12.2”
13:40:30/disk_record=on:eg063h_ys_no0001
....
13:42:30/disk_record=off


recorder
Field System
Schedule:
13:40:10/antcn=17h33m21.2s,16°49’12.2”
13:40:30/disk_record=on:eg063h_ys_no0001
....
13:42:30/disk_record=off


Field System
recorder


Field System
control.exe


Field System
TCP/IP
control.exe

VSI/S - V(LBI) S(tandard) I(nterface) / S(oftware)

- ASCII based
- every **query** or **command** yields **reply**

https://vlbi.org/wp-content/uploads/2023/08/VSI_2003_02_13_vsi-s_final_rev_1.pdf

VSI/S - V(LBI) S(tandard) I(nterface) / S(oftware)

- ASCII based
- every **query** or **command** yields **reply**

https://vlbi.org/wp-content/uploads/2023/08/VSI_2003_02_13_vsi-s_final_rev_1.pdf

```
<command> = <param1> [ : <param2> : ... ];

<query> ? [ <param1> : <param2> : ... ];
```

```
<command> = <param1> [ : <param2> : ... ];
  !<command> = <result code> [: <value1> : ... ];

<query> ? [ <param1> : <param2> : ... ];
  !<query> ? <result code> [: <value1> : ... ];
```


```
mode = mark4 : 64;
```

Field System

control.exe

```
mode = mark4 : 64;
```

```
<< execute
   command >>
```


```
mode = mark4 : 64;
```

```
<< execute
   command >>
```

```
!mode = 0 ;
```

VSI/S - V(LBI) S(tandard) I(nterface) / S(oftware)

<result code> is an ASCII integer as follows:

0 - action successfully completed
1 - action initiated or enabled, but not completed
2 - command not implemented or not relevant to this DTS
3 - syntax error
4 - error encountered during attempt to execute command
5 - currently unable to service request; try again later
6 - inconsistent or conflicting request
7 - no such keyword
8 - parameter error

See pp.8-10 in https://vlbi.org/wp-content/uploads/2023/08/VSI_2003_02_13_vsi-s_final_rev_1.pdf

Different hardware
=
different command set

_rev2?	p. 37	
otect	p. 43	Remove erase protection for
cover	p. 46	Recover data which was overwritten or terminat
eset	p. 48	Reset Mark 5 unit (command only)
SS_rev1?	p. 56	Get StreamStor firmware/software revision le
SS_rev2?	p. 57	Get StreamStor firmware/software revision le
status?	p. 59	Get system status (query only)
task_ID	p. 60	Set task ID (primarily for correlator use)

8.2 Record/Play

mode	p. 31	Set data recording/playback mode
play	p. 38	Play data from current or specified play p
play_rate	p. 40	Set playback data rate; set tvg rate
record	p. 44	Turn recording on|off; assign scan label
scan_play	p. 52	Play scan specified by current value of
scan_set	p. 53	Set scan for scan_check, s
skip	p. 55	Skip forward|backward s

8.3 Data Checking

data_check?	p. 15	Check data starting at p
scan_check?	p. 50	Get scan parameters (q
track_check?	p. 61	Check data on selected
track_set	p. 63	Select tracks for monit

8.4 Data Transfer

disk2file	p. 22	Transfer data from Mark 5 to f
disk2net	p. 24	Transfer data from Mark 5 to
file2disk	p. 28	Transfer data from file to Ma
in2net	p. 30	Transfer data directly from M
net_protocol	p. 35	Set network data-transfer pr
net2disk	p. 33	Transfer data from network
net2out	p. 34	Transfer data directly from

SS_rev?	•	p. 49	Get StreamStor firmware/software revision levels (qu

9.2 System Setup and Monitoring

	Common to Mk5A DIM		
1pps_source		p. 12	Select source of 1pps synchronization tick
clock_set		p. 16	Specify frequency and source of the CLOCK dri
DOT?		p. 27	Get DOT (Data Observe Time) clock informati
DOT_inc		p. 28	Increment DOT clock
DOT_set		p. 29	Set DOT clock on next external 1pps tick
error?	•	p. 31	Get error number/message (query only)
mode	•	p. 35	Set data recording/readback mode
status?	•	p. 51	Get system status (query only)

9.3 Data Checking

disk2net	•	p. 25	Transfer data between start-s
file2disk	•	p. 32	Transfer data from file to M
in2net	•	p. 34	Transfer data directly from
net_protocol	•	p. 37	Set network data-transfer
net2disk	•	p. 36	Transfer data from netw
record	•	p. 41	Turn recording on|off;

OS_rev?	p. 17	Get system information (query only)
protect	p. 24	Get details of operating system (query only)
recover	p. 28	Set/remove erase protection for active module
reset	p. 31	Recover record pointer which was reset abnormally during recording
SS_rev?	p. 32	Reset Mark 5 unit (command only)
	p. 37	Get StreamStor firmware/software revision levels (query only)

7.2 System Setup and Monitoring

error?	p. 18	Get error number/message (query only)
mode	p. 22	Set
status?		cording/readback mode
		tatus (query only)
		uality of the application. Specifies where to store data to, file, disk modul
		MAC addresses to accept (Not supported yet)
		or 10G NIC configuration (Not supported yet)
	p. 35	arting at position of start-scan pointer (query only)
		Check data between start-scan and stop-scan pointers (query only)
		Set start-scan and stop-scan pointers

7.4 Data Transfer

net_protocol	p. 23	Set network data-transfer protocol
record	p. 29	Turn recording on|off; assign scan label
fill_pattern	p. 20	Set StreamStor 32 bit fill pattern
packet	p. 29	Set packet acceptance criteria

Contemporary ethernet recorders

The only tangible difference between the systems. The rest is semantics/software.

Mark6 removable disk packs

FlexBuff fixed disks

High speed packet capture

This does not come for free …

O/S defaults wrong for this use ca
- O/S network buffer sizes t att
- spread (interrupt) over >> cores
- pays no att to hardware layout

Without tuning get < 4 Gbps lossless …

High speed packet capture

Tuning is a topic of its own

Origin of the differences

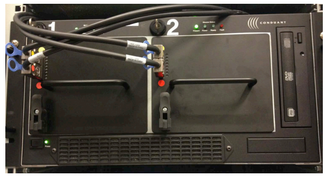
Choice of recording software

Origin of the differences

Choice of recording software

cplane / dplane

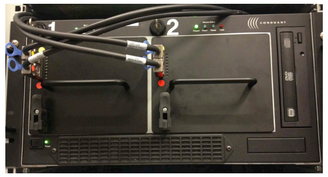
- MIT Haystack



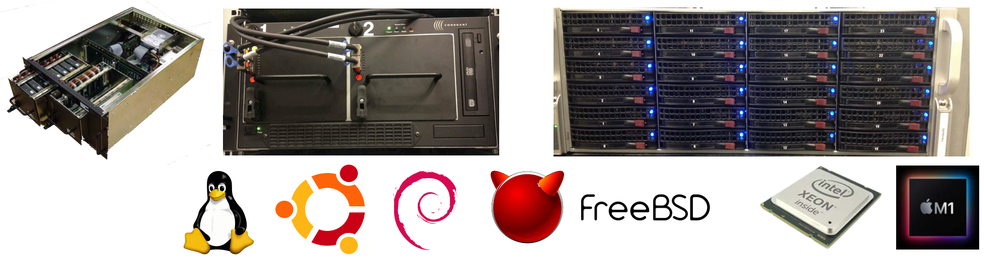
Origin of the differences

Choice of recording software

cplane / dplane
- MIT Haystack



jive5ab(*)
- Joint Institute for VLBI in Europe



…

(*) https://github.com/jive-vlbi/jive5ab

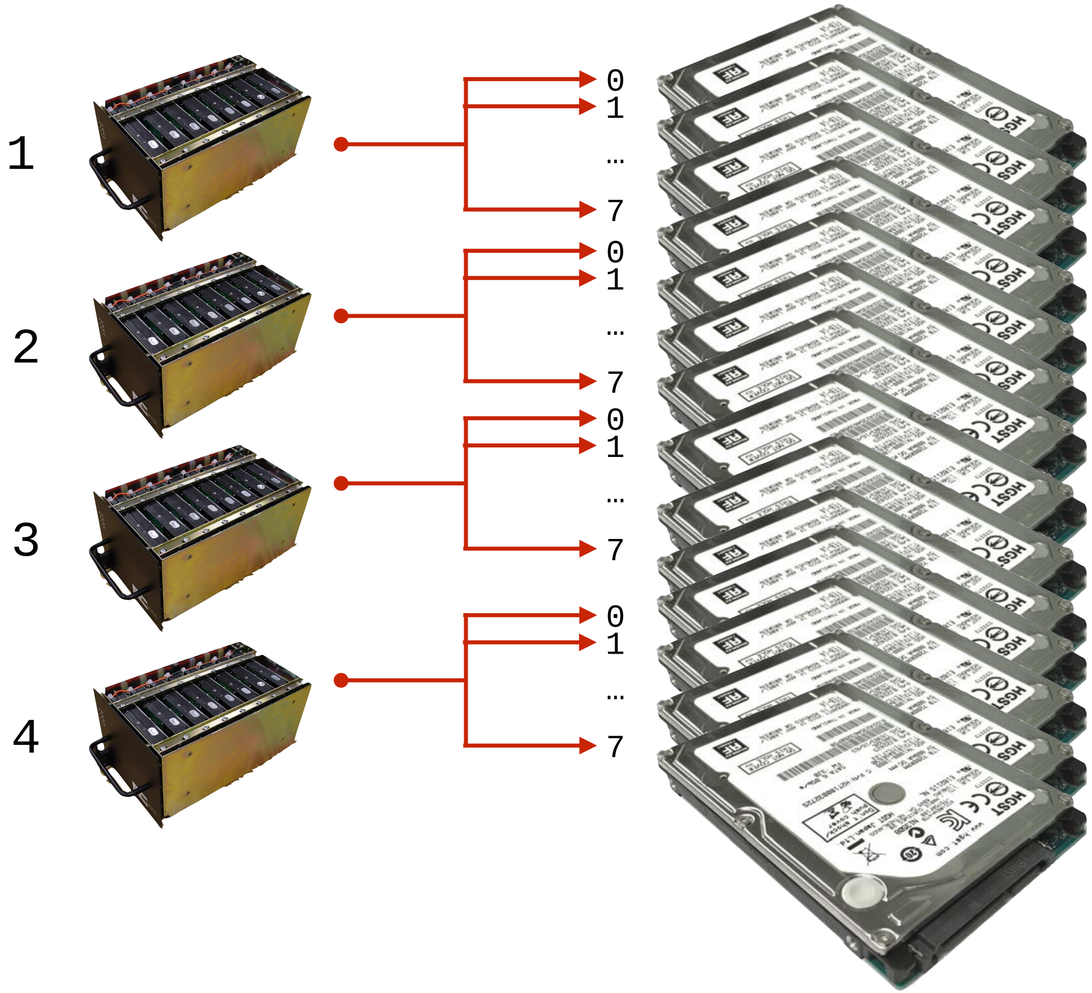
Consequence(s) of the choice

Consequence(s) of the choice

Mount points for the data disks

Consequence(s) of the choice

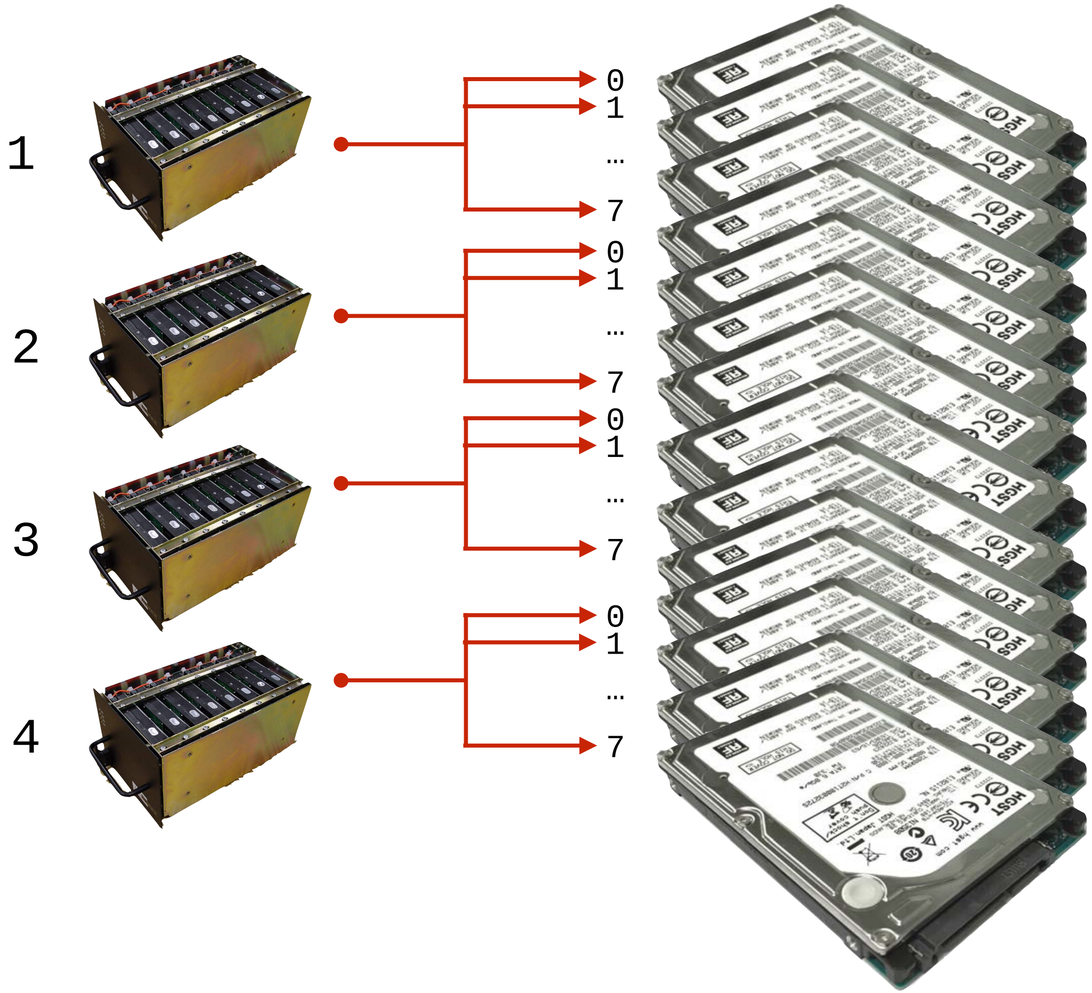
Mount points for the data disks (cplane, Mark6)



Consequence(s) of the choice

Mount points for the data disks (cplane, Mark6)

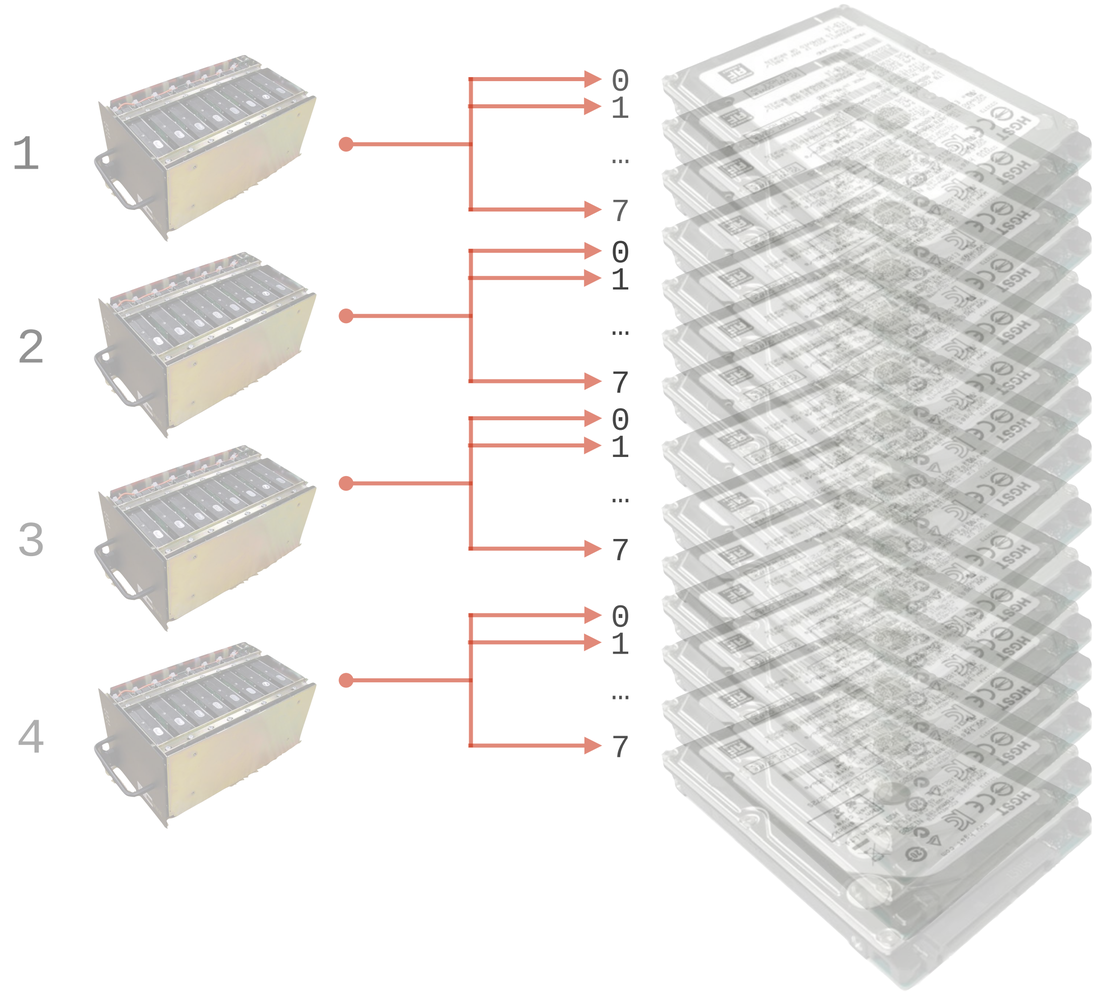
```
/mnt/disk/1/0/data
          /1/1/data
             …
          /1/7/data
/mnt/disk/2/0/data
             …
          /2/7/data
       …
/mnt/disk/4/7/data
```



Consequence(s) of the choice

Mount points for the data disks (cplane, Mark6)

/mnt/disk/**1/0**/data
/**1/1**/data
…
/**1/7**/data
/mnt/disk/**2/0**/data
…
/**2/7**/data
…
/mnt/disk/**4/7**/data



regex: /mnt/disk/[1-4]/[0-7]/data

Consequence(s) of the choice

Mount points for the data disks (jive5ab, *)

/mnt/… ???
/data/… ???
/… ???

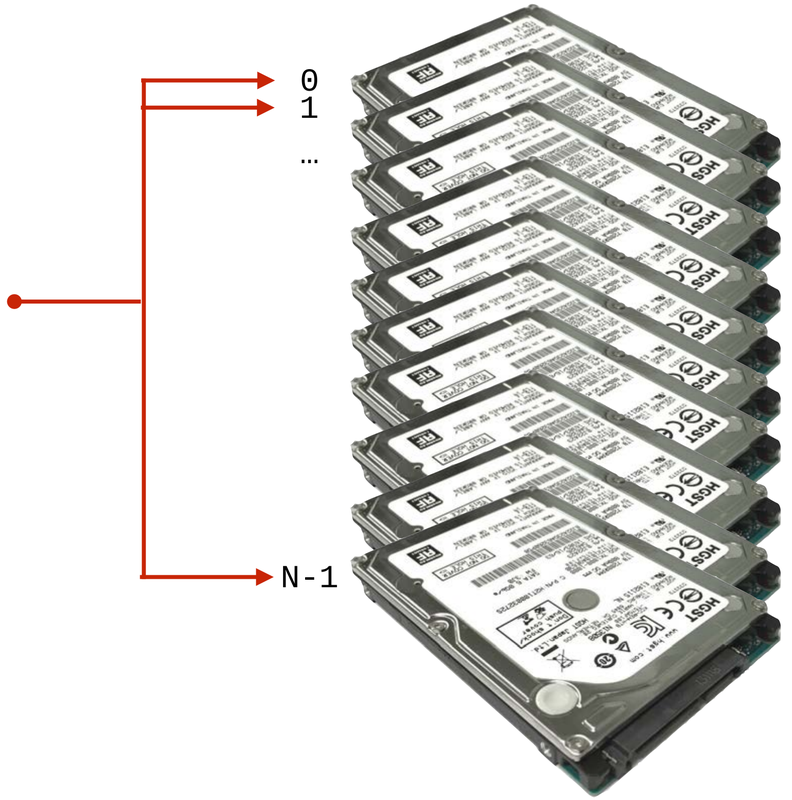
jive5ab doesn't care,
BUT …

Consequence(s) of the choice

Mount points for the data disks (jive5ab, DEFAULT startup)

/mnt/disk*0*
/mnt/disk*1*
…
/mnt/disk*N-1*

jive5ab looks for those at startup
does NOT mount them!

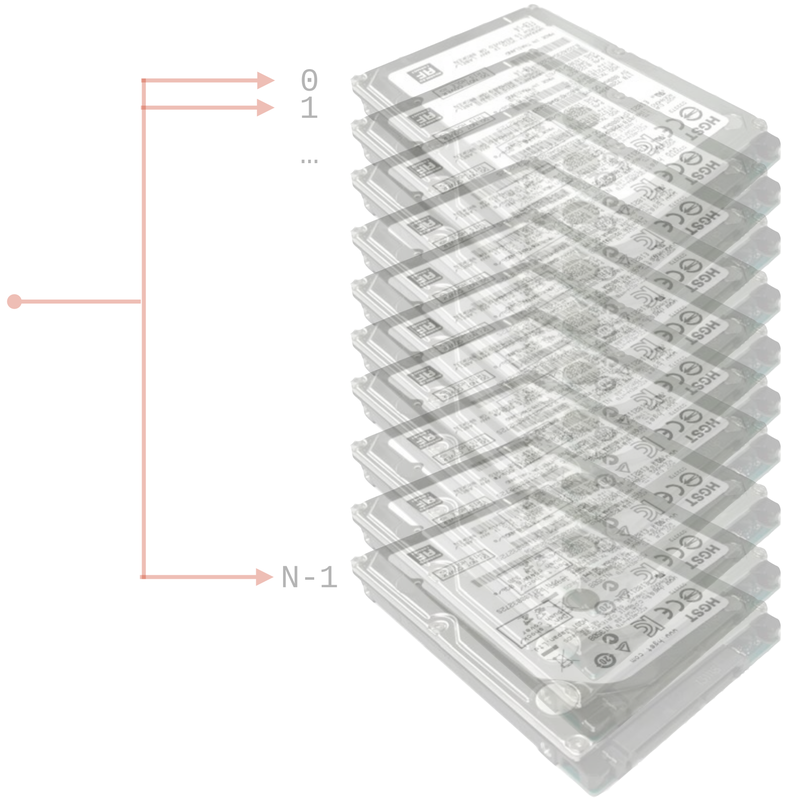


Consequence(s) of the choice

Mount points for the data disks (jive5ab, DEFAULT startup)

/mnt/disk*0*
/mnt/disk*1*
…
/mnt/disk*N-1*

jive5ab looks for those at startup
does NOT mount them!



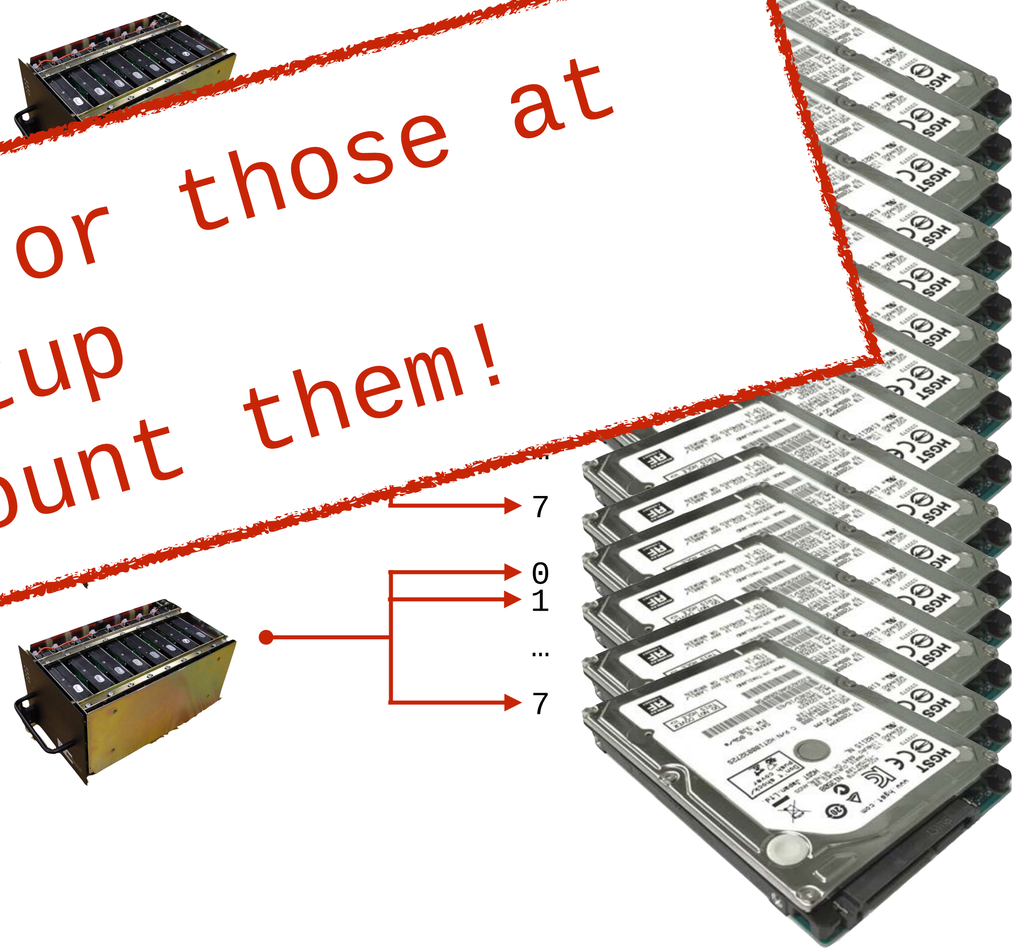
regex: /mnt/disk[0-9]+

Consequence(s) of the choice

Mount points for the data disks (jive5ab, **-6** command line option)

/mnt/disk/**1/0**/data
/**1/1**/data
…
/**1/7**/d
/mnt/dis
/mnt/dis

jive5ab looks for those at startup
does NOT mount them!



Consequence(s) of the choice

Mount points for the data disks (jive5ab, *)

```
set_disks?; (*)
⇒ !set_disks? 0 : /mnt/disk0 : /mnt/disk1 : … ;

set_disks = 12 : /path/to/sd* ; (*)
⇒ !set_disks = 0 : 18 ;

set_disks?; (*)
⇒ !set_disks? 0 : /mnt/disk/1/0/data : … : /path/to/sdA : … ;
```

(*) https://github.com/jive-vlbi/jive5ab/blob/master/doc/jive5ab-documentation-1.11.pdf

Consequence(s) of the choice

Mount points for the data disks (jive5ab, command line script)

```
$> m6sg_mount
```

(*) https://github.com/jive-vlbi/jive5ab/blob/master/scripts/m6sg_mount

Consequence(s) of the choice

Mount points for the data disks (jive5ab, command line script)

1 

```
$> m6sg_mount 134
```

3 

4 

(*) https://github.com/jive-vlbi/jive5ab/blob/master/scripts/m6sg_mount

Consequence(s) of the choice

Mount points for the data disks (jive5ab, command line script)

1 

```
$> m6sg_mount -u 3
```

3 

4 

(*) https://github.com/jive-vlbi/jive5ab/blob/master/scripts/m6sg_mount

Consequence(s) of the choice

How the packets are read from the network

Consequence(s) of the choice

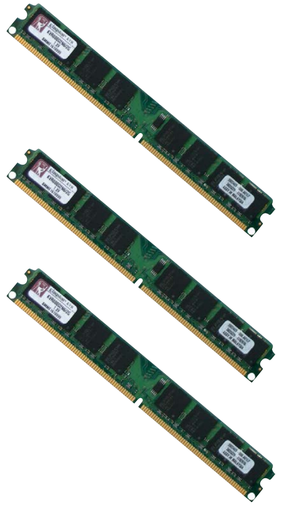
How the packets are read from the network (cplane / dplane)

/dev/eth**1**

$> tcpdump -c 10M /dev/eth**1**

/dev/eth**N**

$> tcpdump -c 10M /dev/eth**N**



10 MByte blocks

Consequence(s) of the choice

How the packets are read from the network (cplane / dplane)

/dev/eth**1**

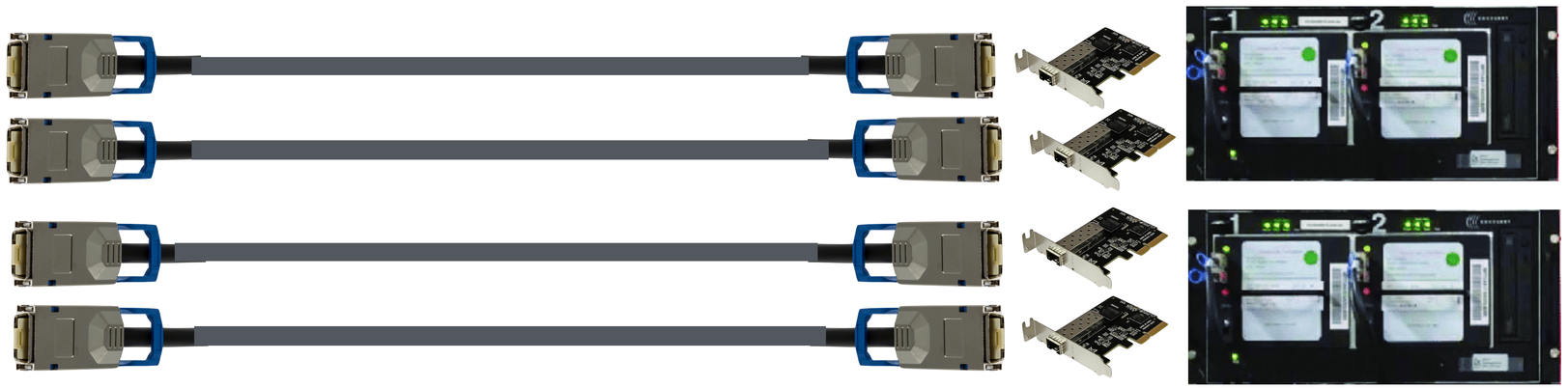
$> tcpdump

No valid network configuration necessary on the network cards or digital back end!

10 MByte blocks

Consequence(s) of the choice

How the packets are read from the network (cplane / dplane)



Configuration: ☺

Flexibility: ☹

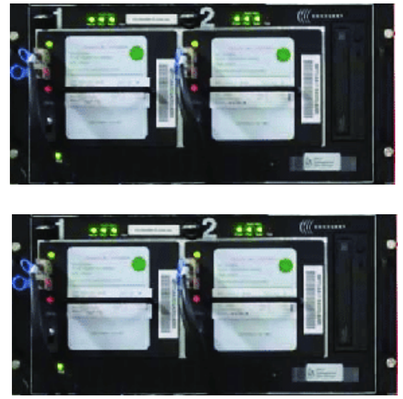
Consequence(s) of the choice

How the packets are read from the network (cplane / dplane)

BAND A, R+L

BAND B, R+L

R+R



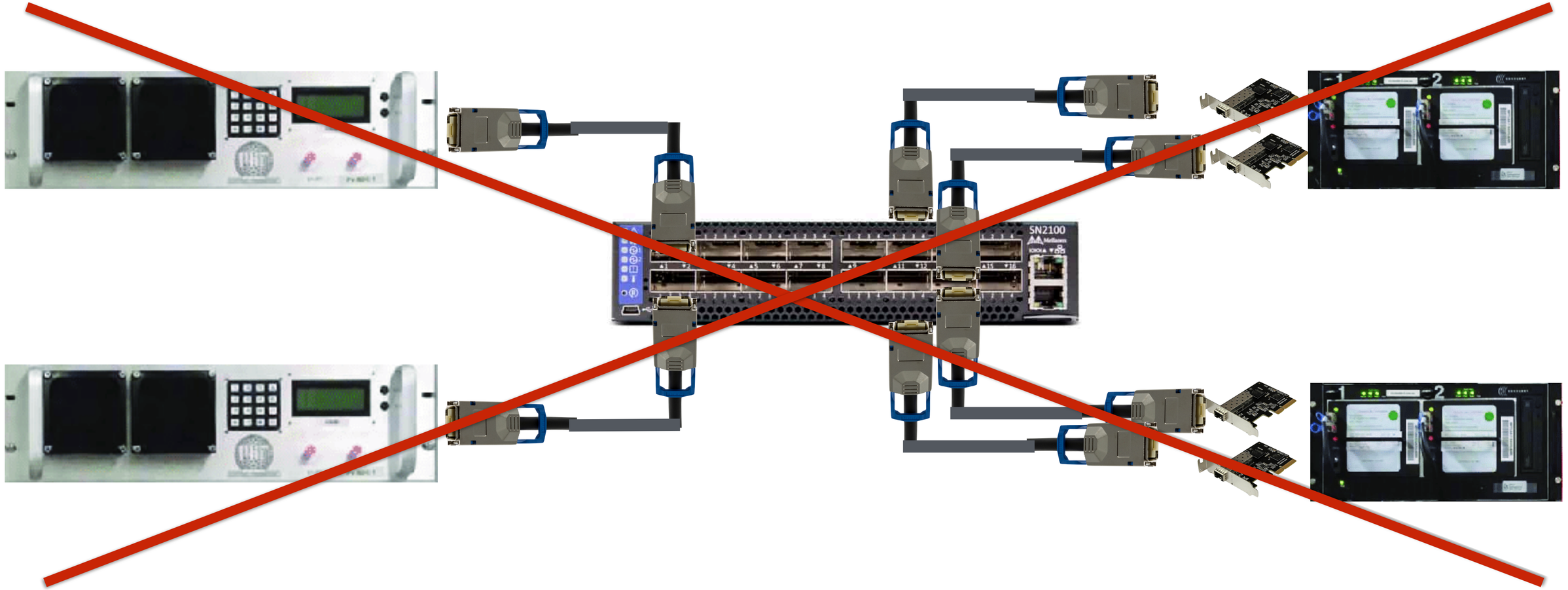
L+L

Installation: ☺

Flexibility: ☹

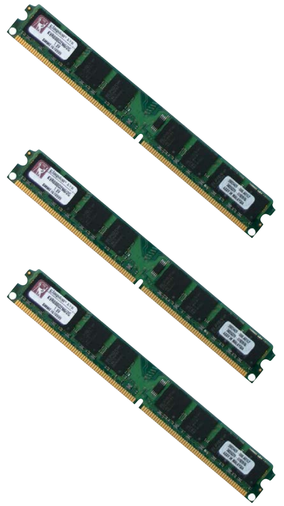
Consequence(s) of the choice

How the packets are read from the network (cplane / dplane)



Consequence(s) of the choice

How the packets are read from the network (jive5ab)



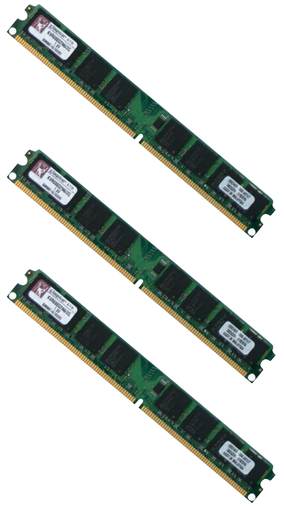
Consequence(s) of the choice

How the packets are read from the network (jive5ab)

192.168.178.0

192.168.178.1

```
s = socket(AF_INET, PF_DGRAM);
bind(s, IPv4Address:PORT_NR);

while( true ) {
    read(socket, buf, N);
    /* … */
}
```



N Byte blocks

Consequence(s) of the choice

How the packets are read from the network (jive5ab)

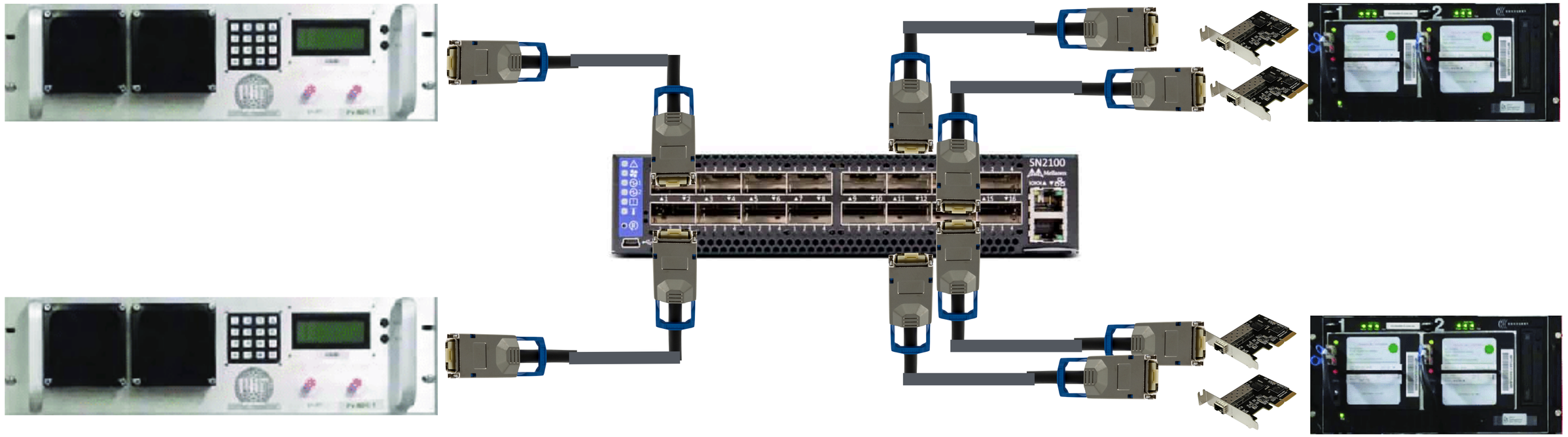
192.168.178.0

REQUIRE valid network configuration on the network cards and digital back ends!

N Byte blocks

Consequence(s) of the choice

How the packets are read from the network (jive5ab)



Configuration: 😕

Flexibility: 🙂

Consequence(s) of the choice

What gets written to disk

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

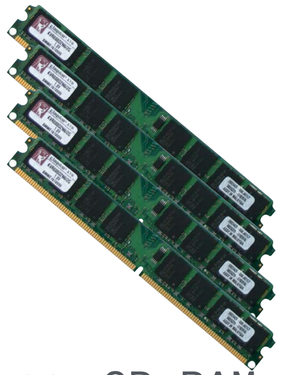
HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR



>> GB RAM

Recording application header

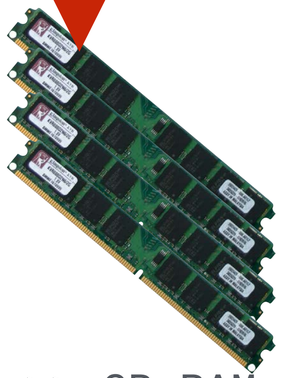
VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards

10 MByte



>> GB RAM

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR | 0

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

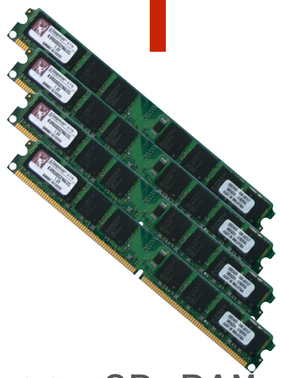
HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR



>> GB RAM

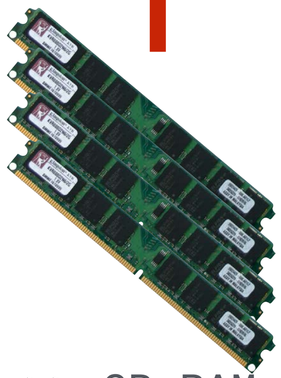
Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards



>> GB RAM

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR | 0 | DATA

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR

Recording application header

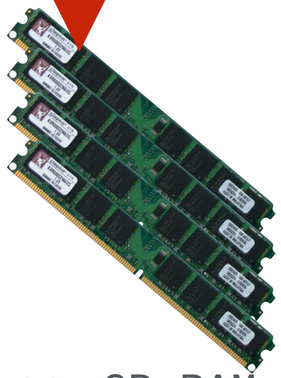
VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards

10 MByte



>> GB RAM

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR | 0 | DATA

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR

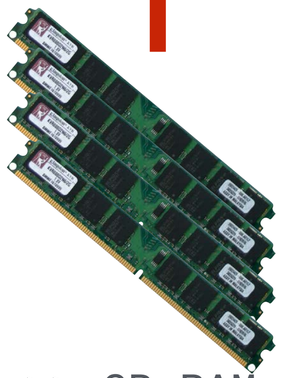
Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards



>> GB RAM

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR | 0 | DATA

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

HDR

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

HDR | 1 | DATA

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

ethernet cards

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

HDR	0	DATA	7	DATA

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

HDR	2	DATA	4	DATA

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

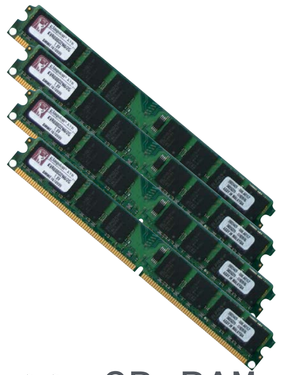
HDR	1	DATA	5	DATA

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

HDR	3	DATA	6	DATA

Recording application header

VLBI payload data



>> GB RAM

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

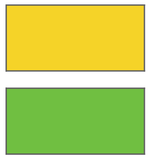
/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

Inconvenient for e-shipping



Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (cplane / dplane Mark6 format)

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

VDIF content cannot be used directly

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

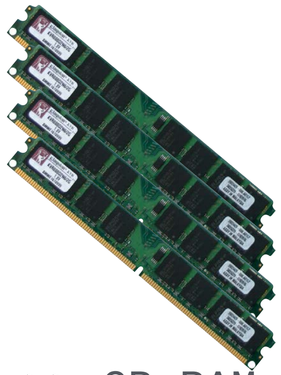
ethernet cards

/mnt/disk0/**EXP_STATION_SCAN/**

/mnt/disk1/**EXP_STATION_SCAN/**

/mnt/disk2/**EXP_STATION_SCAN/**

/mnt/disk*N*/**EXP_STATION_SCAN/**



>> GB RAM

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

ethernet cards

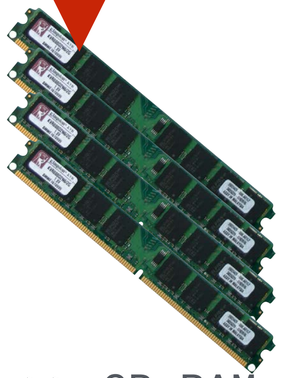
N byte

/mnt/disk0/**EXP_STATION_SCAN/**

/mnt/disk1/**EXP_STATION_SCAN/**

/mnt/disk2/**EXP_STATION_SCAN/**

/mnt/disk*N*/**EXP_STATION_SCAN/**



>> GB RAM

Recording application header

VLBI payload data

N = 256 MB typically

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

ethernet cards

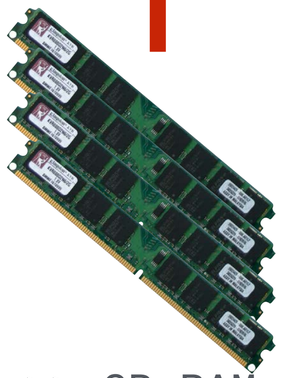
/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

/mnt/disk1/**EXP_STATION_SCAN/**

/mnt/disk2/**EXP_STATION_SCAN/**

/mnt/disk$N$/**EXP_STATION_SCAN/**



>> GB RAM

Recording application header

VLBI payload data

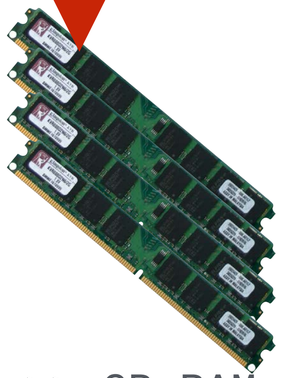
N = 256 MB typically

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

ethernet cards

N byte



>> GB RAM

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

/mnt/disk1/**EXP_STATION_SCAN/**

/mnt/disk2/**EXP_STATION_SCAN/**

/mnt/disk*N*/**EXP_STATION_SCAN/**

Recording application header

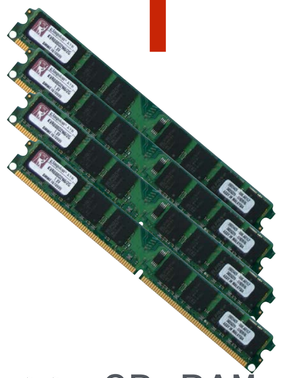
VLBI payload data

N = 256 MB typically

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

ethernet cards



>> GB RAM

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

/mnt/disk1/**EXP_STATION_SCAN/**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

/mnt/disk$N$/**EXP_STATION_SCAN/**

Recording application header

VLBI payload data

N = 256 MB typically

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

ethernet cards

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

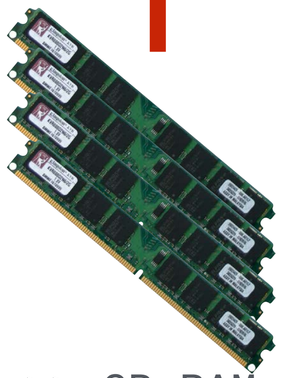
EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk$N$/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**



>> GB RAM

Recording application header

VLBI payload data

N = 256 MB typically

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk*N*/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

Unique file names: e-transfer ✓

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (jive5ab, FlexBuff/vbs format (DEFAULT))

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk*N*/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

Only VLBI data: can use any VDIF tool

Recording application header

VLBI payload data

Consequence(s) of the choice

What gets written to disk (jive5ab - you have a choice)

compiled in default:

vbs (FlexBuff) format

command line: set default format

```
$> jive5ab --format mk6|flexbuff
```

runtime: set format (VSI/S)

```
record = mk6 : 0|1 ;
```

FlexBuff how-to

1. buy/repurpose machine
2. install+tune operating system (any POSIX)
3. connect (many) disks
4. can mount as /mnt/diskNNN?
   - yes: done
   - no: remember path/regex
     in FS jive5ab.ctl add set_disks= path/regex;
5. configure network card(s)
6. get + build jive5ab

```
$> git clone https://github.com/jive-vlbi/jive5ab.git
$> mkdir build && cd build && cmake -DSSAPI_ROOT=nossapi ..
$> make [-j <n_cpu>] [VERBOSE=1]
```

7. profit! $> jive5ab -m3 [options]

Tuning

the key to high speed packet capture

CPU/core limitations

Achieving *very* high capture rates

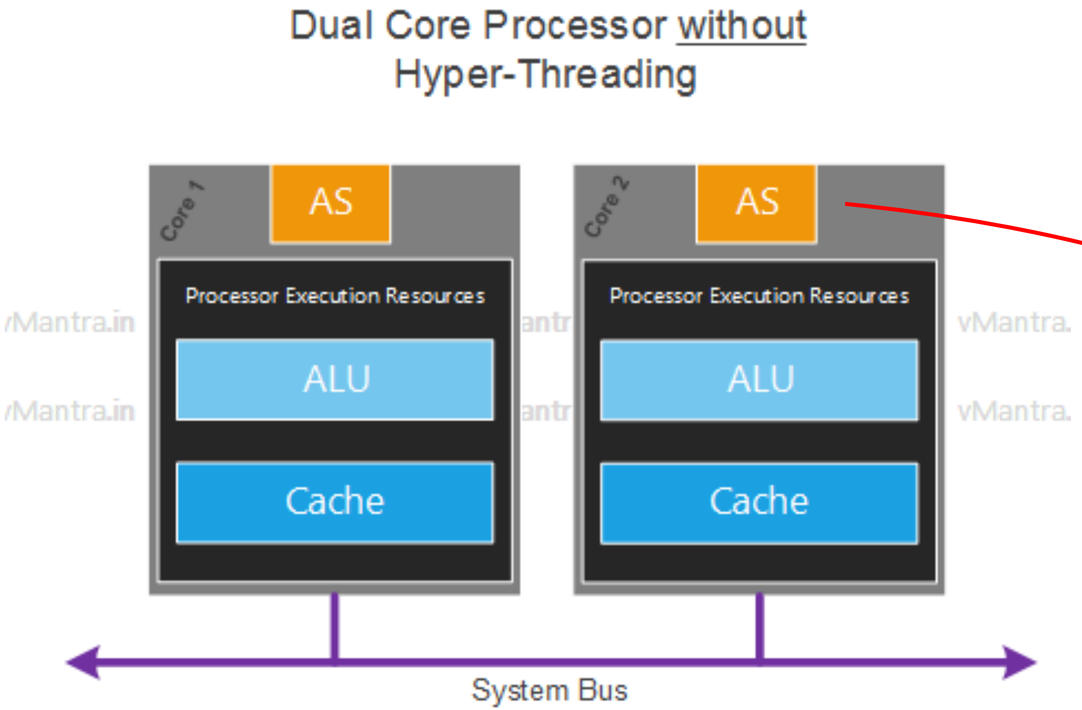
NO HYPERTHREADING

CPU/core limitations

Achieving *very* high capture rates

CPU/core limitations

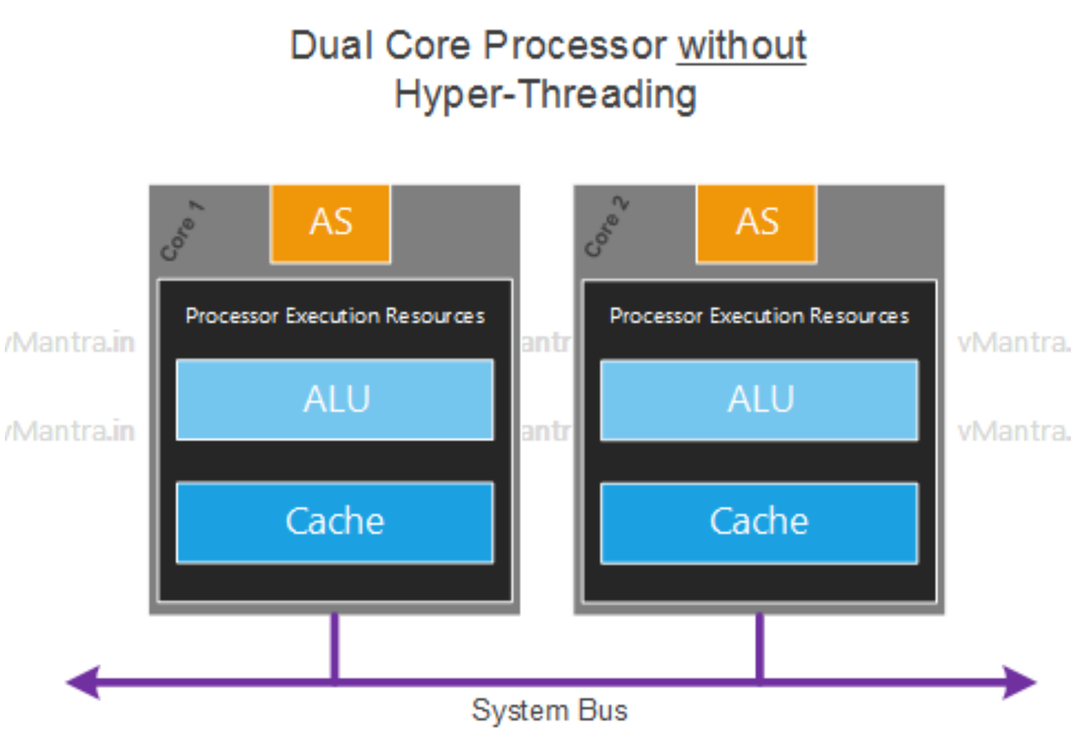
Achieving *very* high capture rates



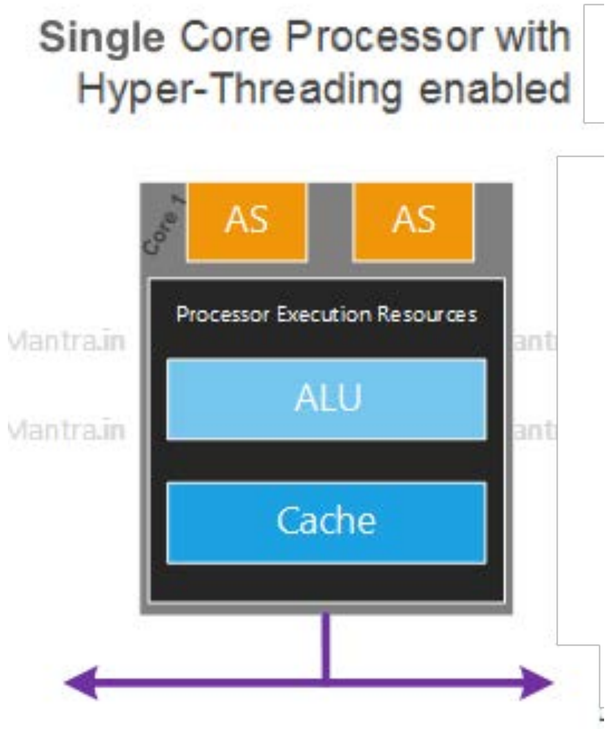
AS = Architecture State
which means
"CPU registers and flags &cet."

CPU/core limitations

Achieving *very* high capture rates

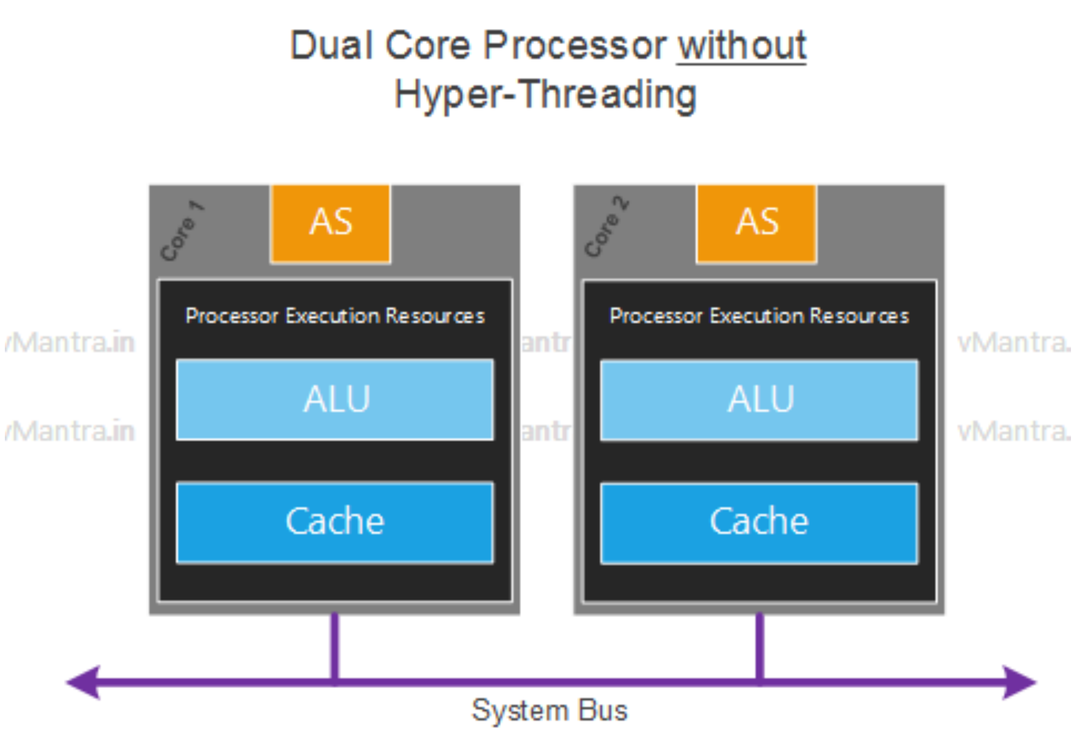


VS.

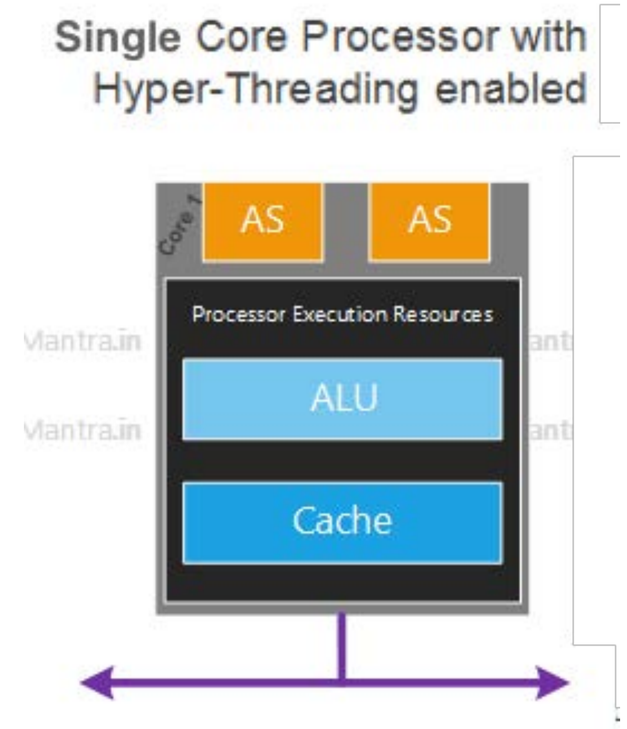


CPU/core limitations

Achieving *very* high capture rates

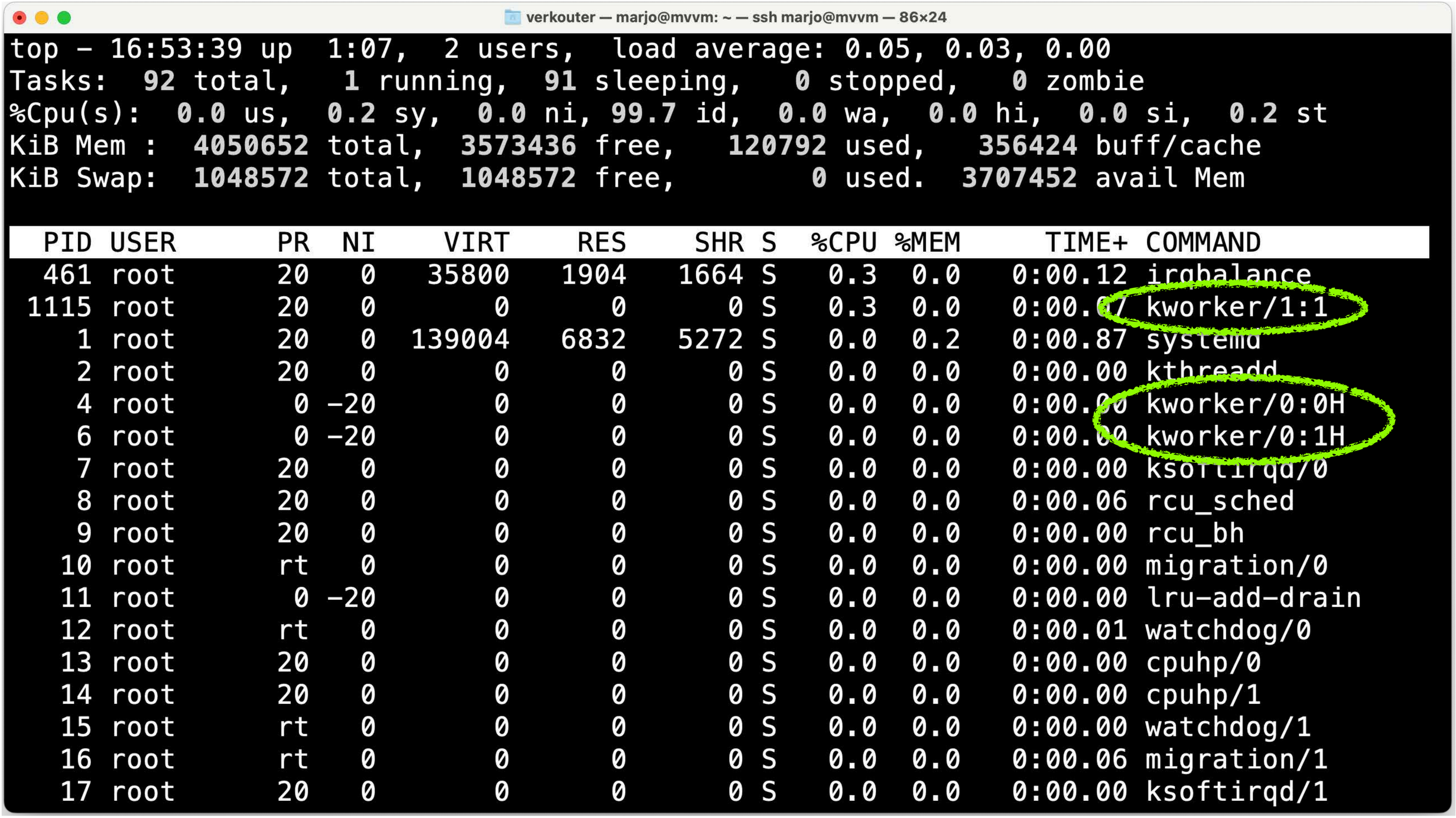


VS.



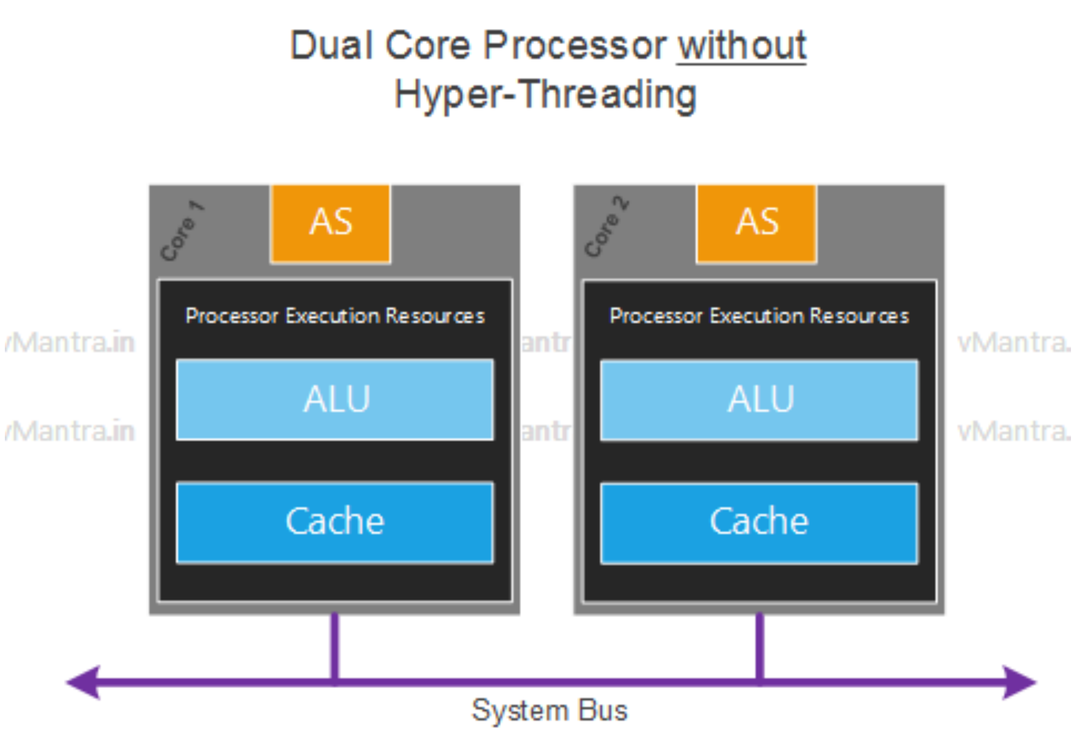

```
$> grep -E '^processor' /proc/cpuinfo | wc -l
2
```

```
$> grep -E '^processor' /proc/cpuinfo | wc -l
2
```

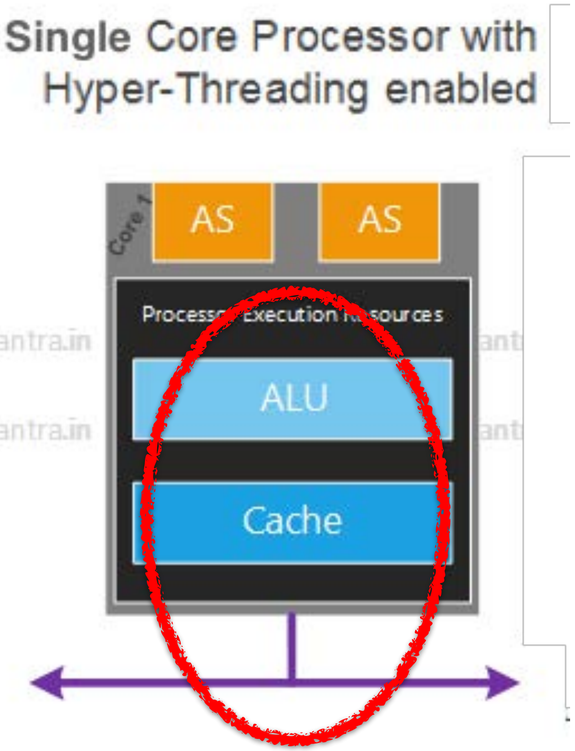

verkouter — marjo@mvvm: ~ — ssh marjo@mvvm — 86×24
top - 16:53:39 up 1:07, 2 users, load average: 0.05, 0.03, 0.00
Tasks: 92 total,   1 running, 91 sleeping,   0 stopped,   0 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.2 st
KiB Mem : 4050652 total, 3573436 free,   120792 used,   356424 buff/cache
KiB Swap: 1048572 total, 1048572 free,   0 used. 3707452 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
461 root 20 0 35800 1904 1664 S 0.3 0.0 0:00.12 irqbalance
1115 root 20 0 0 0 0 S 0.3 0.0 0:00.07 kworker/1:1
1 root 20 0 139004 6832 5272 S 0.0 0.2 0:00.87 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
4 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
6 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:1H
7 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
8 root 20 0 0 0 0 S 0.0 0.0 0:00.06 rcu_sched
9 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_bh
10 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
11 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 lru-add-drain
12 root rt 0 0 0 0 S 0.0 0.0 0:00.01 watchdog/0
13 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
15 root rt 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/1
16 root rt 0 0 0 0 S 0.0 0.0 0:00.06 migration/1
17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/1

CPU/core limitations

Achieving *very* high capture rates



VS.

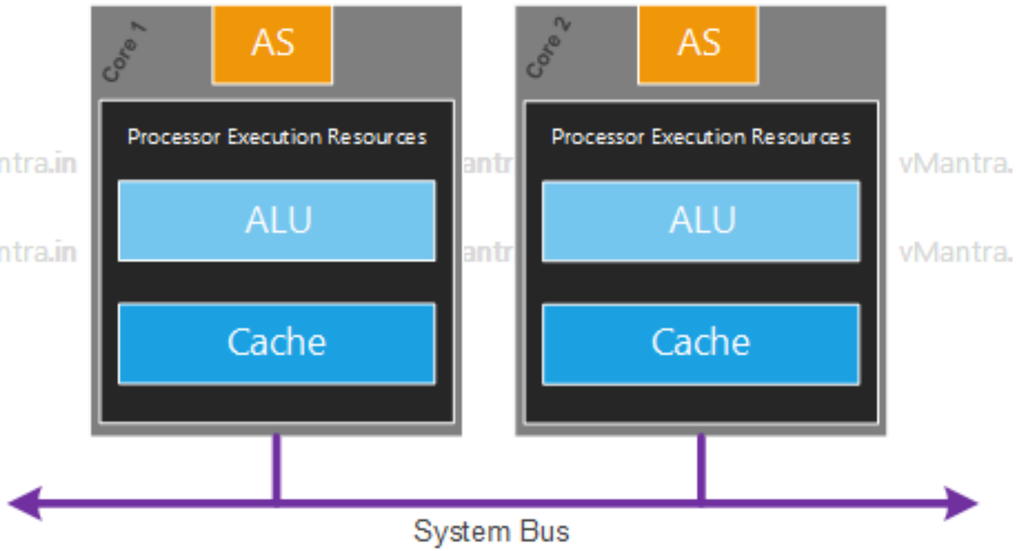


CPU/core limitations

Achieving *very* high capture rates

```
thread0 {
  read(net, …);
}
```

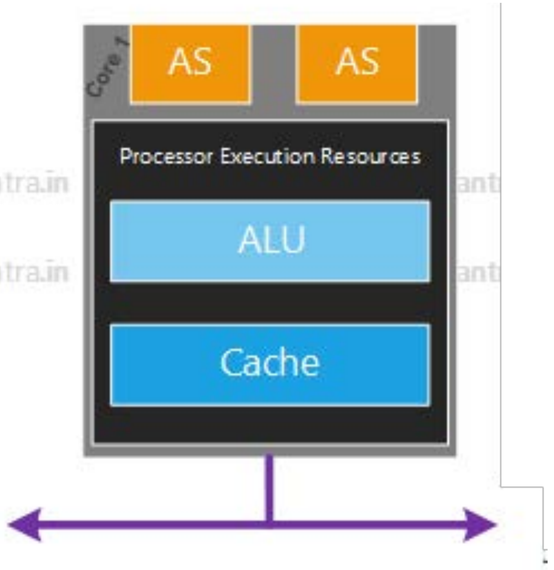
```
thread1 {
  write(disk, …);
}
```



VS.

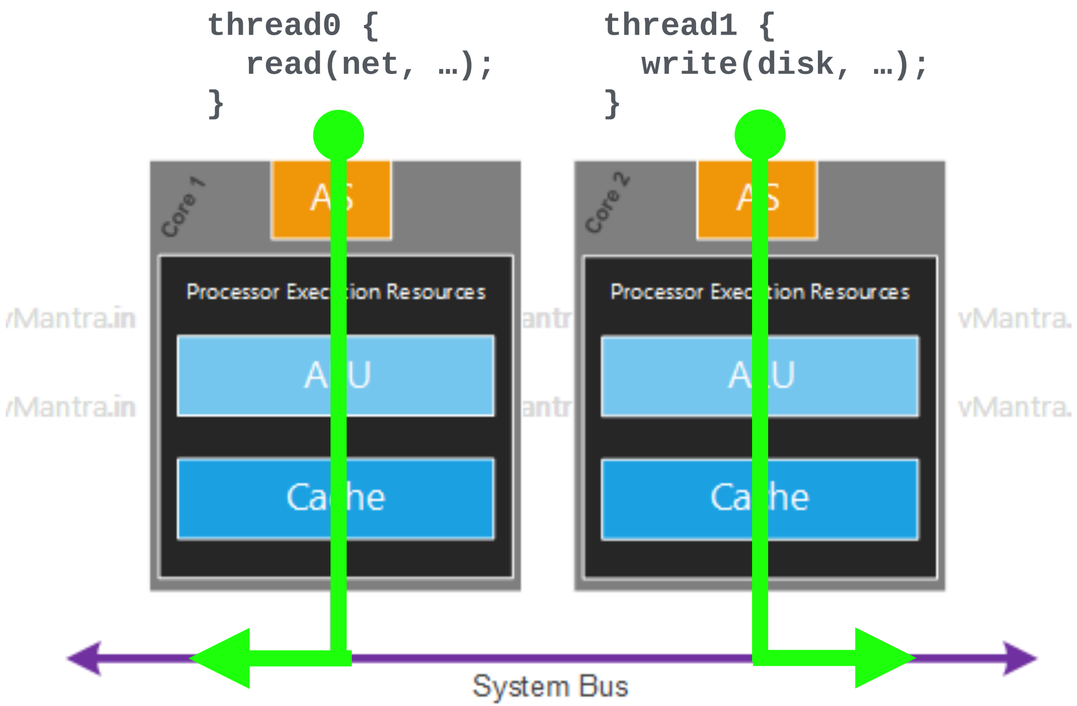
```
thread0 {
  read(net, …);
}
```

```
thread1 {
  write(disk, …);
}
```

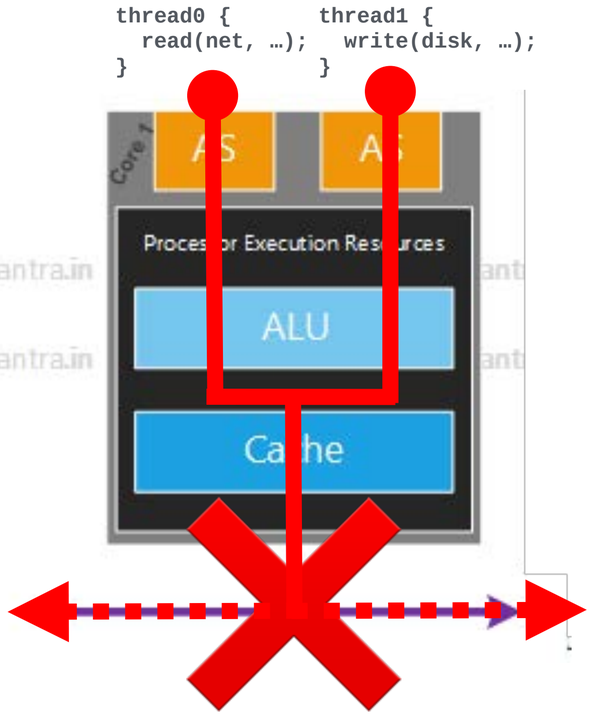


CPU/core limitations

Achieving *very* high capture rates (#1 NO hyperthreading)



VS.



CPU/core limitations

Achieving *very* high capture rates

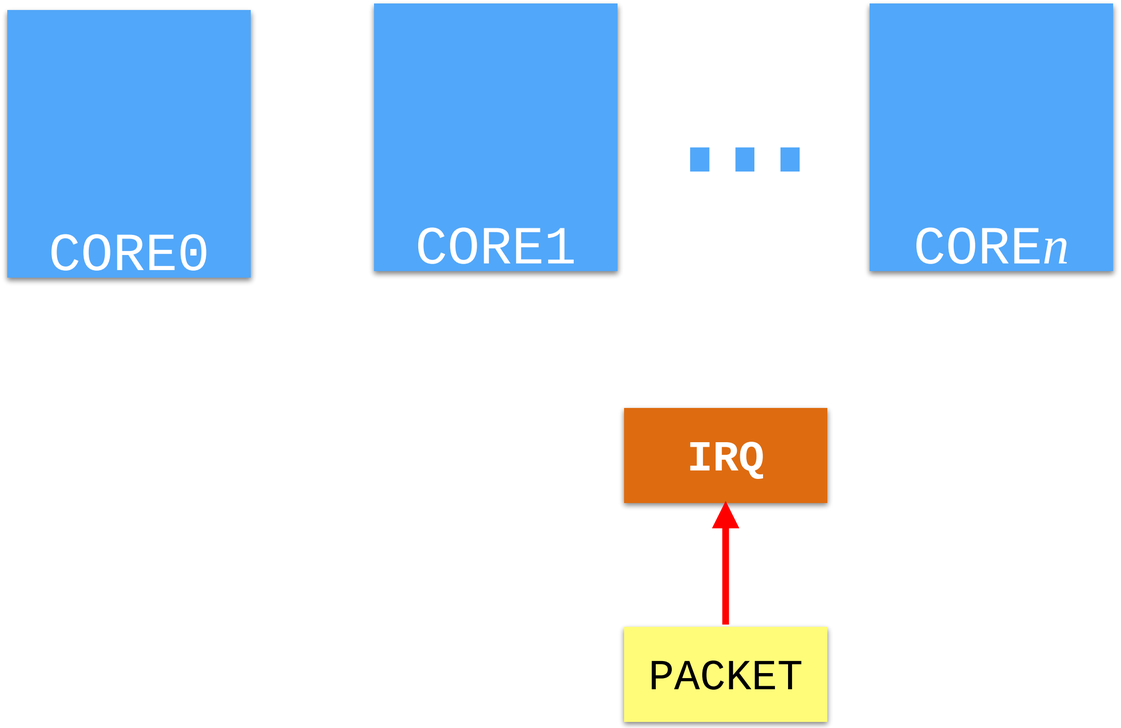
Interrupt pinning

CPU/core limitations

Achieving *very* high capture rates

CPU/core limitations

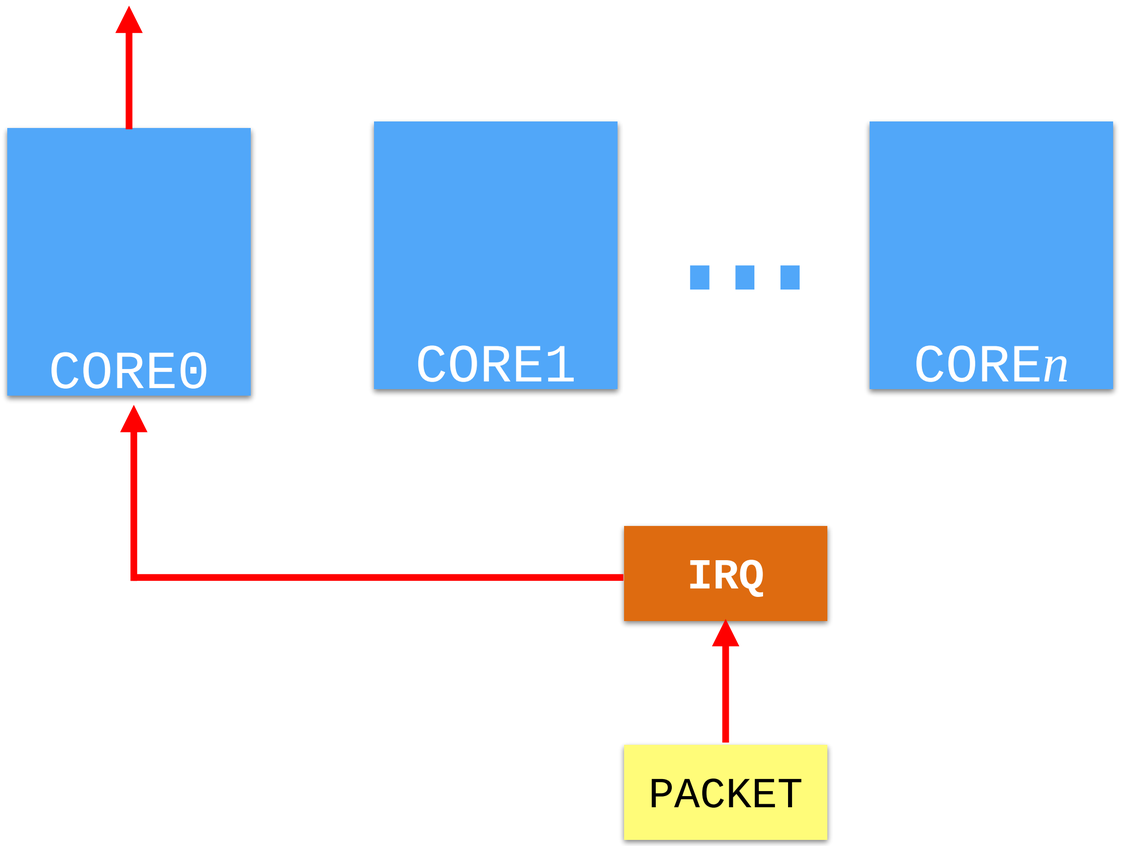
Achieving *very* high capture rates



CPU/core limitations

Achieving *very* high capture rates

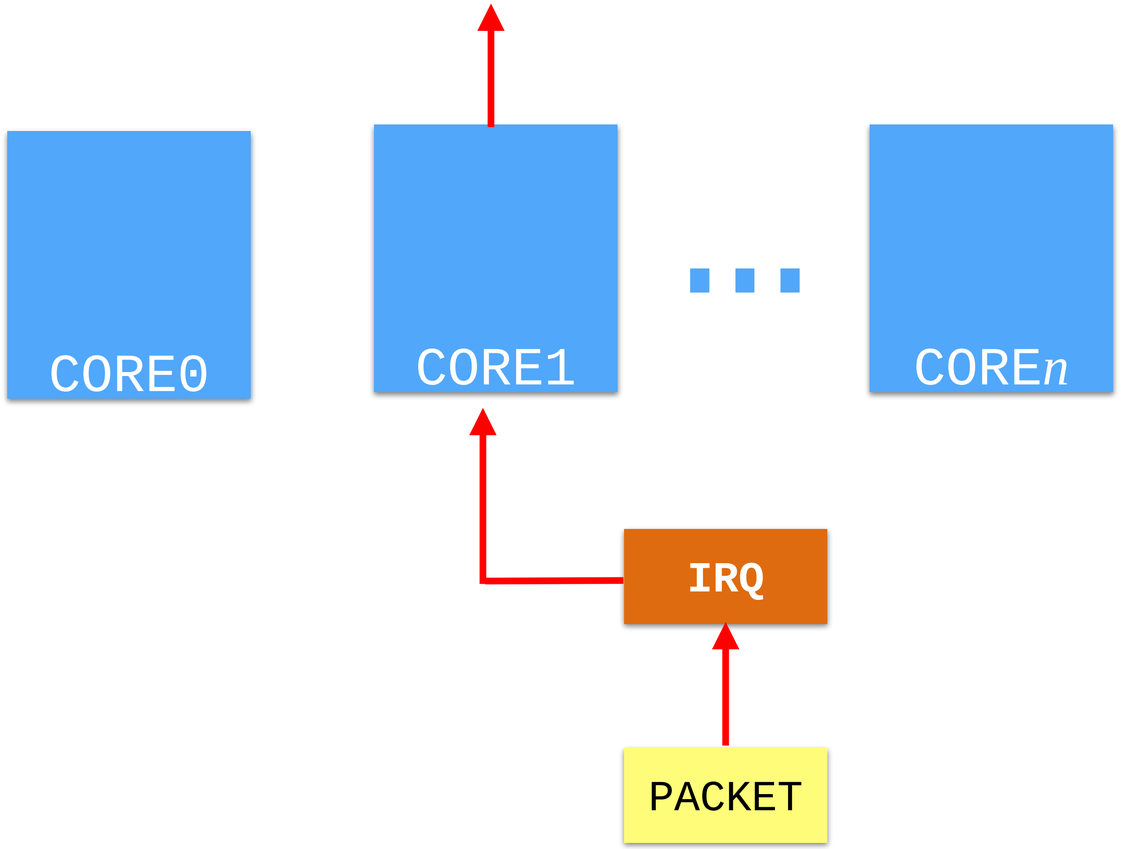
```
char* p;
if(*p & 0x1){
   ...
}
```



CPU/core limitations

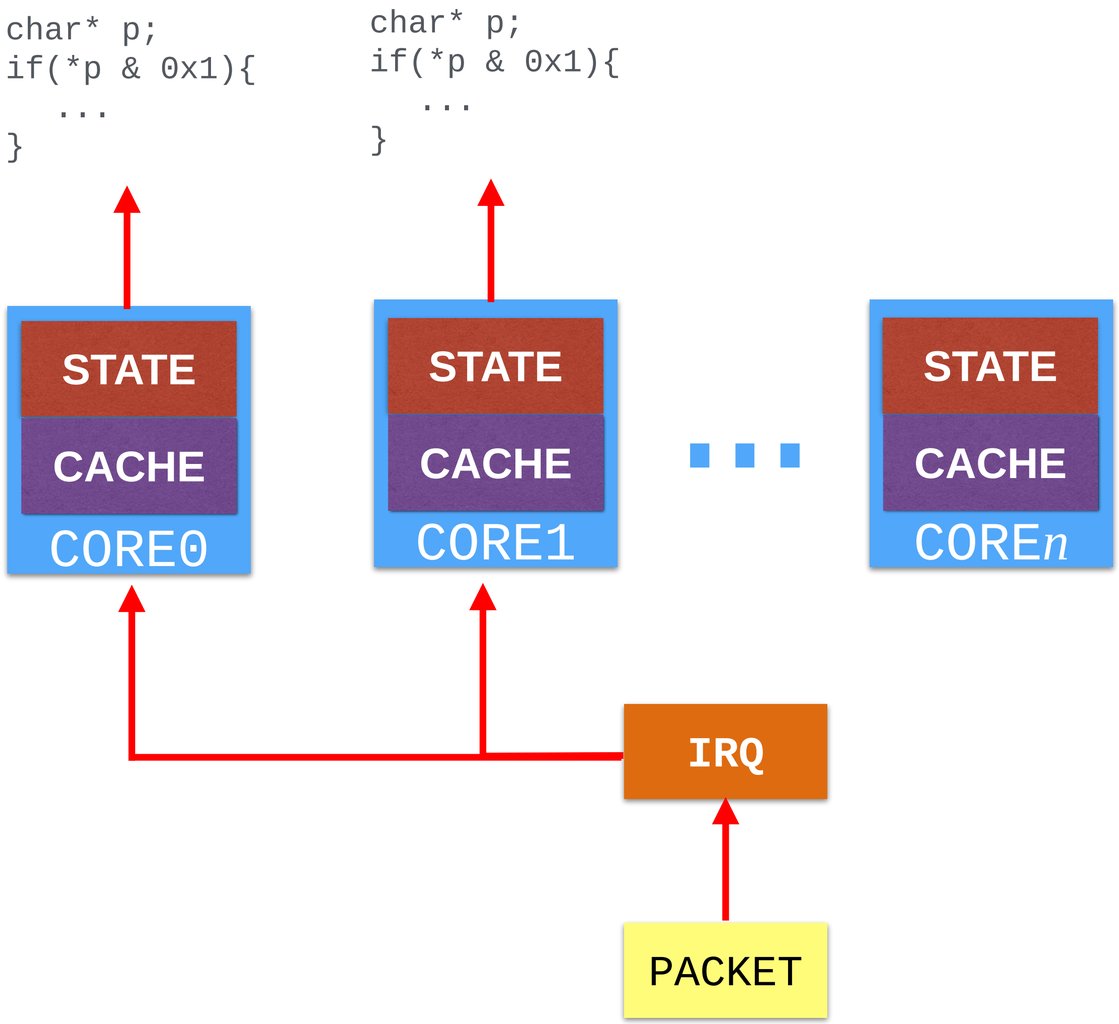
Achieving *very* high capture rates

```
char* p;
if(*p & 0x1){
   ...
}
```



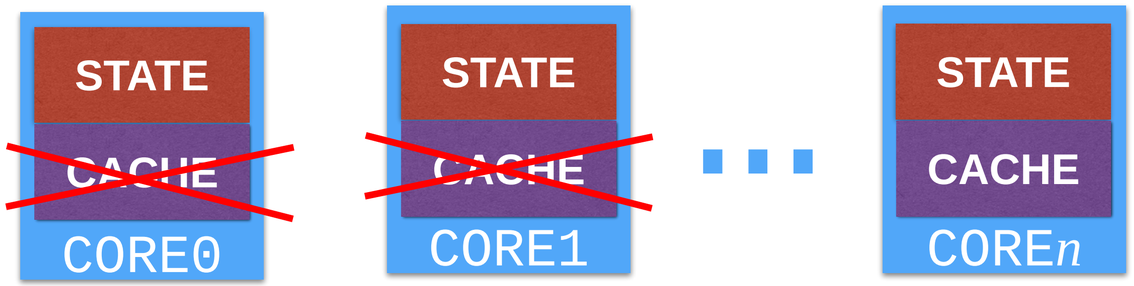
CPU/core limitations

Achieving *very* high capture rates



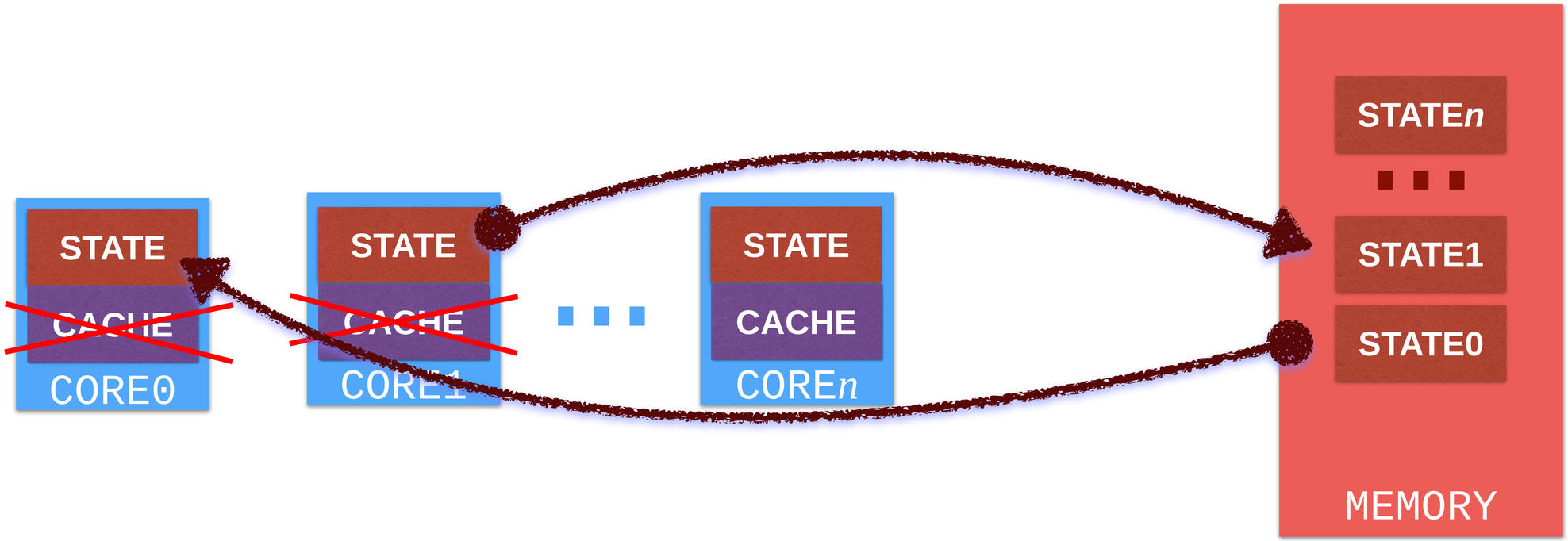
CPU/core limitations

Achieving *very* high capture rates



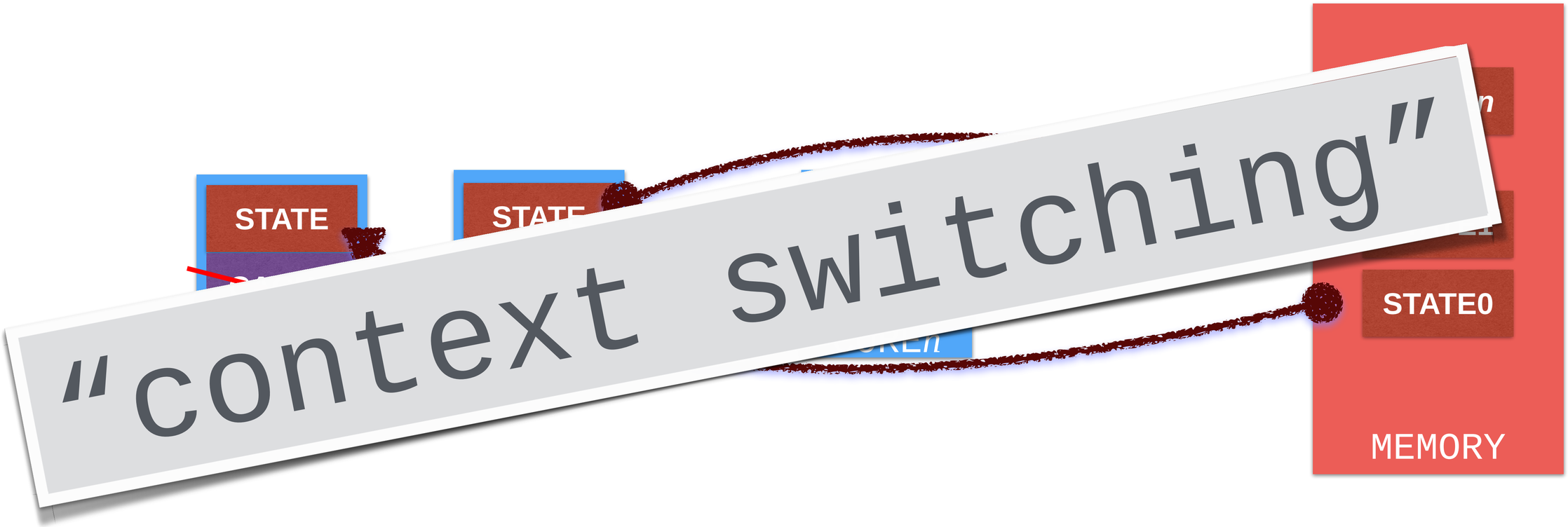
CPU/core limitations

Achieving *very* high capture rates



CPU/core limitations

Achieving *very* high capture rates



CPU/core limitations

Achieving *very* high capture rates (#2 IRQ scheduling)

Linux default:

round-robin IRQ scheduling

Solution:

*pin **ethernet** IRQs to fixed core(s)*
to keep data local to core + cache

CPU/core limitations

Achieving *very* high capture rates

Buffer sizes

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep udp
net.ipv4.udp_mem        = 92031  122711  184062
net.ipv4.udp_rmem_min = 4096
net.ipv4.udp_wmem_min = 4096
net.core.rmem_max = 1073741824
<< snip >>
```

https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep udp
net.ipv4.udp_mem        = 92031   122711   184062
net.ipv4.udp_rmem_min = 4096
net.ipv4.udp_wmem_min = 4096
<< snip >>
```

~370 MB Number of pages allowed for queueing by all UDP sockets

https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep -E 'net.core.[rw]mem'
net.core.rmem_default = 212992
net.core.rmem_max = 212992
net.core.wmem_default = 212992
net.core.wmem_max = 212992
```

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep -E 'net.core.[rw]mem'
net.core.rmem_default = 212992
net.core.rmem_max = 212992
net.core.wmem_default = 212992
net.core.wmem_max = 212992
```

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep backlog
net.core.netdev_max_backlog = 1000
net.ipv4.tcp_max_syn_backlog = 128
```

CPU/core limitations

Achieving *very* high capture rates

```
$ sudo /sbin/sysctl -a | grep backlog
net.core.netdev_max_backlog = 1000
net.ipv4.tcp_max_syn_backlog = 128
```

CPU/core limitations

Achieving *very* high capture rates (#3 Tune buffers)

```
$ cat tune.txt(*)
net.core.rmem_max=201326592
net.ipv4.udp_mem="536870912 805306368 1073741824”
net.core.netdev_max_backlog=1048576
<< snip: there is more! >>

$ sudo /sbin/sysctl -ptune.txt
```

(*) See https://github.com/jive-vlbi/jive5ab/doc/flexbuf.recording.txt (in 3.1.0-branch and later)

A documented ‘script’ to serve as tuning guide: the what, why, and how

CPU/core limitations

Achieving *very* high capture rates

Single core performance

CPU/core limitations

Achieving *very* high capture rates

Single core: 16-21* Gbps reliably

- per packet interrupt handling
- memory speed limit

(*) broadly scales with CPU clock frequency

CPU/core limitations

Achieving *very* high capture rates (#4 multiple separate streams)

Create separate ≤ 16 Gbps streams:

- different IP destination address
- different UDP destination port
- or both

CPU/core limitations

Achieving *very* high capture rates (#4 multiple separate streams)

Create separate ≤ 16 [illegible]

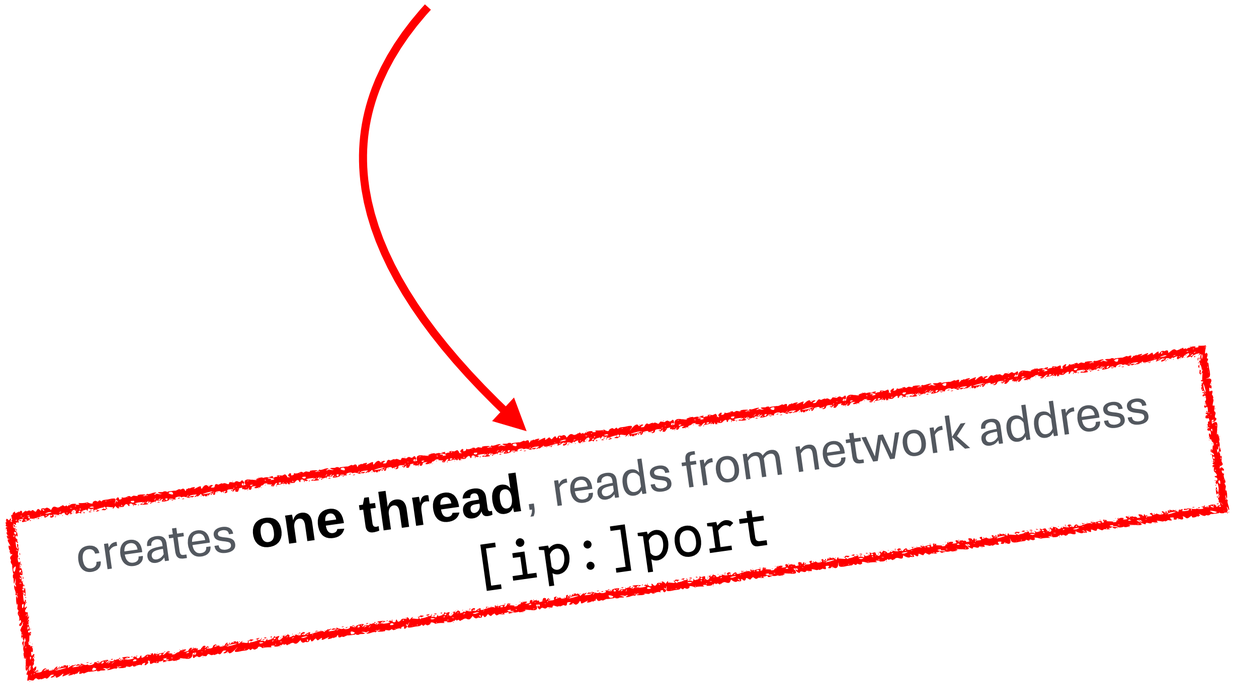
- diff[illegible]

Setting up + starting not really supported

CPU/co…ions

Achieving *very* high capt… (separate streams)

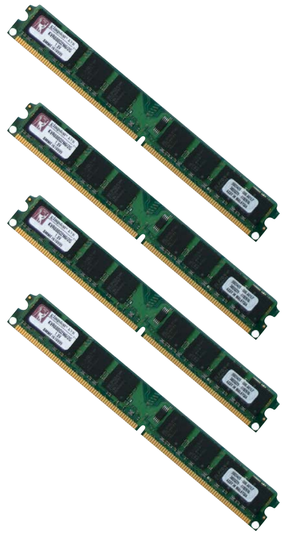
```
net_port = <port> ;
  record = on : <scan> ;
```



ethernet cards

memory modules

hard disks



Network reader thread

```
s = socket(AF_INET, PF_DGRAM);
bind(s, ANY:PORT_NR);

while( true ) {
    read(socket, buf, N);
    /* … */
}
```

Disk writer thread #1

```
fd = open(“/mnt/disk0/…”);

while( t
    writ
    /* …
}
```

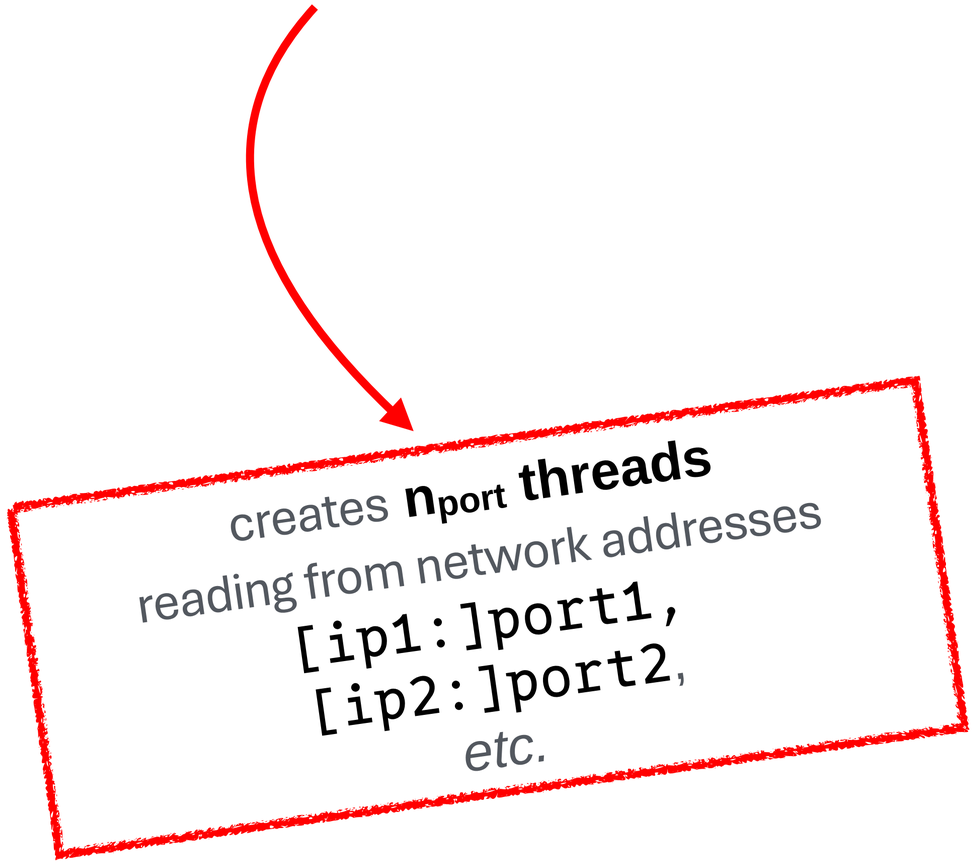
Disk writer thread #2

```
fd = open(“/mnt/disk0/…”);

while( t
    writ
    /* …
}
```

Disk writer thread #3

```
fd = open(“/mnt/disk0/…”);

while( true ) {
    write(fd, buf, N);
    /* … */
}
```

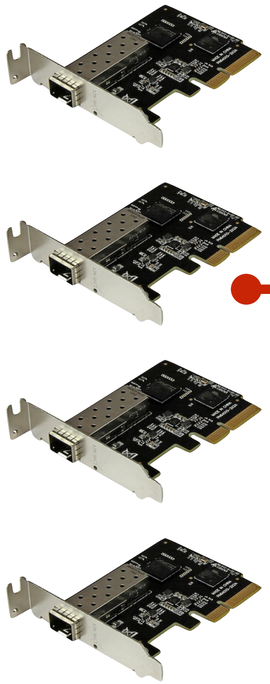
```
net_port = [ip1@]port1 [:[ip2@]port2]* ;
  record = on : <scan> ;
```

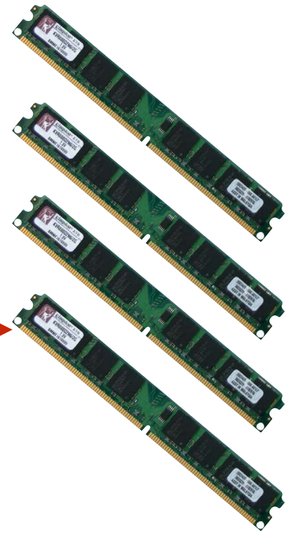


ethernet cards

memory modules

hard disks





```
Network reader thread #1
s = socket(AF_INET, PF_DGRAM);
bind(s
while(
    re
    /*
}
```

```
Network reader thread #2
s = socket(AF_INET, PF_DGRAM);
bind(s,
while( t
    read
    /* …
}
```

```
Network reader thread #3
s = socket(AF_INET, PF_DGRAM);
bind(s, IP3:PORT_NR3);

while( true ) {
    read(socket, buf, N);
    /* … */
}
```

```
Disk writer thread #1
fd = open(“/mnt/disk0/…”);

while( t
    writ
    /* …
}
```

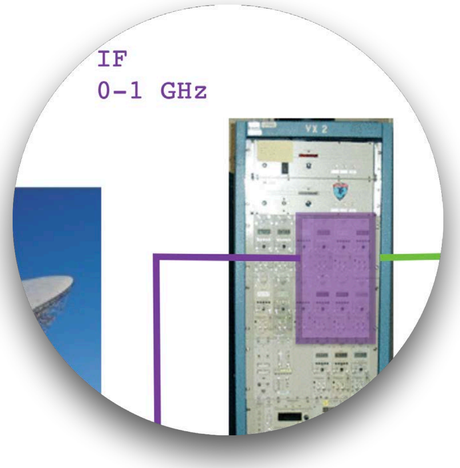
```
Disk writer thread #2
fd = open(“/mnt/disk0/…”);
while( t
    writ
    /* …
}
```

```
Disk writer thread #3
fd = open(“/mnt/disk0/…”);

while( true ) {
    write(fd, buf, N);
    /* … */
}
```

Data format(s)

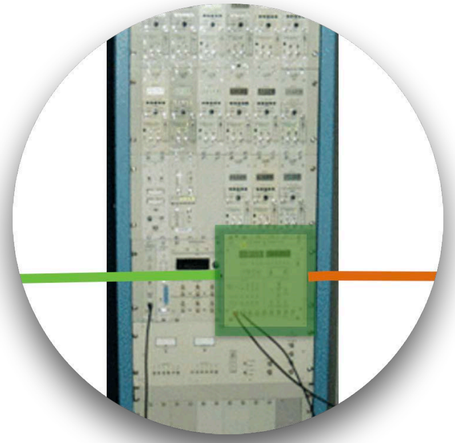
what did I just record?

	t_0	t_1	t_2	t_3	t_4	…									
BBC0$_v$	10	11	01	11	01	10	00	10	10	00	10	01	11	01	10
BBC1$_v$	01	00	10	10	00	11	10	00	10	01	11	01	10	00	01
…															
BBCN_v	00	10	01	11	01	10	00	01	01	11	01	10	00	11	10



	t_0	t_1	t_2	t_3	t_4	…
BBC0$_v$	10	11	01	11	01	10 00 10 10 00 10 01 11 01 10
BBC1$_v$	01	00	10	10	00	11 10 00 10 01 11 01 10 00 01
…						
BBCN_v	00	10	01	11	01	10 00 01 01 11 01 10 00 11 10

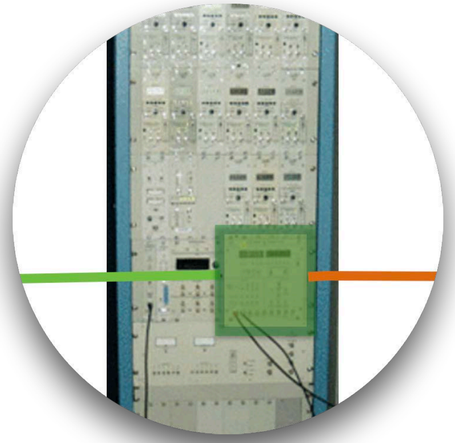
t_0	10 01 … 00
t_1	11 00 … 10



	t_0	t_1	t_2	t_3	t_4	…
BBC0$_v$	10	11	01	11	01	10 00 10 10 00 10 01 11 01 10
BBC1$_v$	01	00	10	10	00	11 10 00 10 01 11 01 10 00 01
…						
BBCN_v	00	10	01	11	01	10 00 01 01 11 01 10 00 11 10

t_0	10 01 … 00
t_1	11 00 … 10
t_2	01 10 … 01
…	

$$\overbrace{n_{channel} \times n_{bits\ per\ sample} \div 8}$$



binary representation of format specific header

t_0 **10 01 … 00**

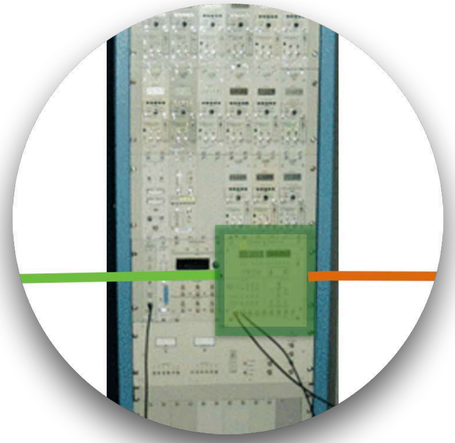
t_1 **11 00 … 10**

11 11 … 11

01 01 … 10

t_2 **01 10 … 01**

…



Mark5B

adb459ed	390dd99b	45f31dd4	9798e953
5d0a9d3e	81dd1a87	98976aae	70a862a2
e76bae65	2e6a79f4	41d589e3	9d6e991e
31349721	aaddd6bc	df83926a	7a78a5eb
abaddeed	bead04b1	65467925	18762097
0595f62c	c6752d53	2c61a994	fa85d6d9
97c4ac8c	db15ebaf	95a519b5	219b6379
a69a1469	fc7859ab	43c77c54	4a158ea6
9d75f2f7	58ea8499	11f9076b	7d8cf754
552619b8	a79b9ba7	d5e7fe86	44b849d5

note: this is in 32-bit hexadecimal representation now

adb459ed	390dd99b	45f31dd4	9798e953
5d0a9d3e	81dd1a87	98976aae	70a862a2
e76bae65	2e6a79f4	41d589e3	9d6e991e
31349721	aaddd6bc	df83926a	7a78a5eb
abaddeed	bead04b1	65467925	18762097
0595f62c	c6752d53	2c61a994	fa85d6d9
97c4ac8c	db15ebaf	95a519b5	219b6379
a69a1469	fc7859ab	43c77c54	4a158ea6
9d75f2f7	58ea8499	11f9076b	7d8cf754
552619b8	a79b9ba7	d5e7fe86	44b849d5

note: this is in 32-bit hexadecimal representation now

VDIF EDV 0

6b5669a1	82c65526	a9b65a1f	63aa656c
54e695f5	9a8d9857	58975895	a7498aa2
2dd189e6	8999ac5b	d964f565	2269c9f6
9ae5296b	a8875542	4b87557d	
0a37956f			
			0000000
	662db168	2979f057	56959798
56256a76	e4ad879b	6bc1d467	c6d23918
94a4ad83	69d2665e	f625c279	6a12b1ab
46cb9389	ada02966	25694a99	a5a3975b
e6e46da8	61ba145a	72669a17	95931b29
dcd9e5a2	1d396981	b492a659	b194a6c5

no magic pattern!

6b5669a1	82c65526	a9b65a1f	63aa656c
54e695f5	9a8d9857	58975895	a7498aa2
2dd189e6	8999ac5b	d964f565	2269c9f6
9ae5296b	a8875542	4b87557d	5197668a
0a37956f	**2200066a**	**000003ec**	**4005069**
00000000	**00000000**	**00000000**	**0000000**
62ec9736	662db168	2979f057	56959798
56256a76	e4ad879b	6bc1d467	c6d23918
94a4ad83	69d2665e	f625c279	6a12b1ab
46cb9389	ada02966	25694a99	a5a3975b
e6e46da8	61ba145a	72669a17	95931b29
dcd9e5a2	1d396981	b492a659	b194a6c5

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

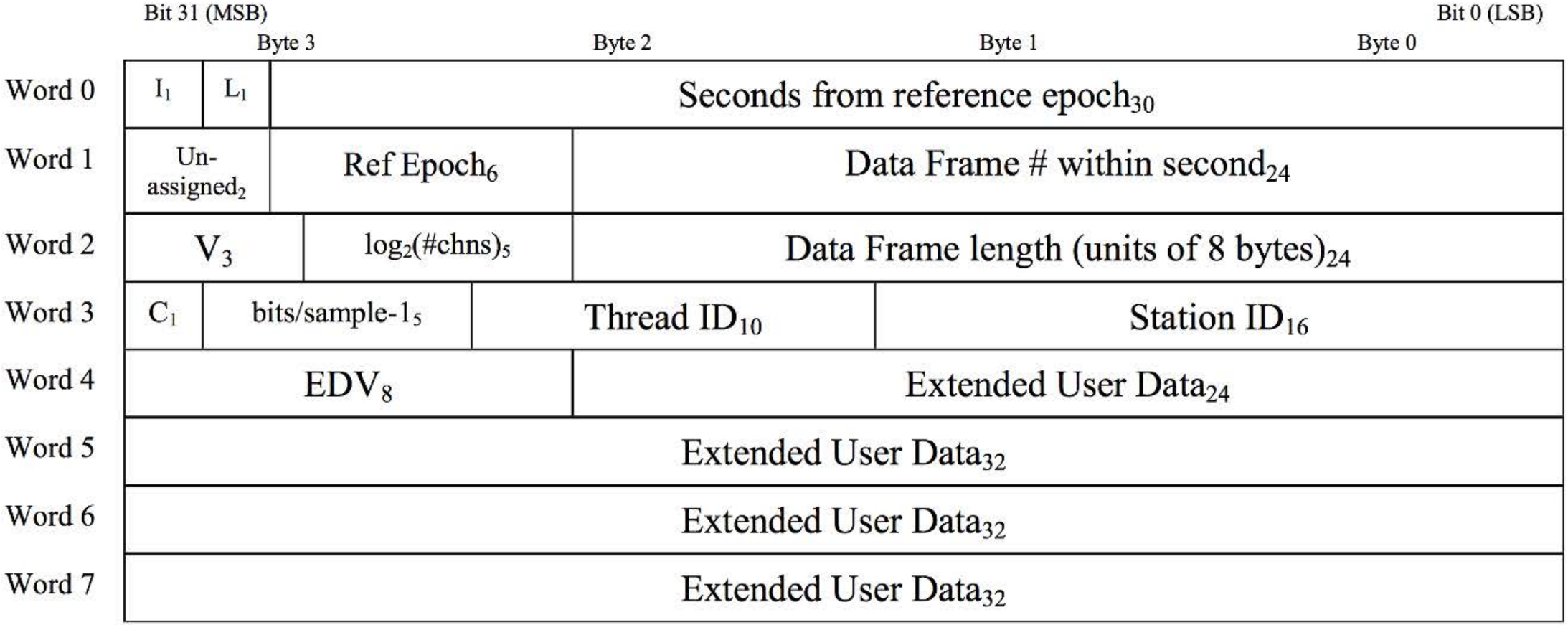
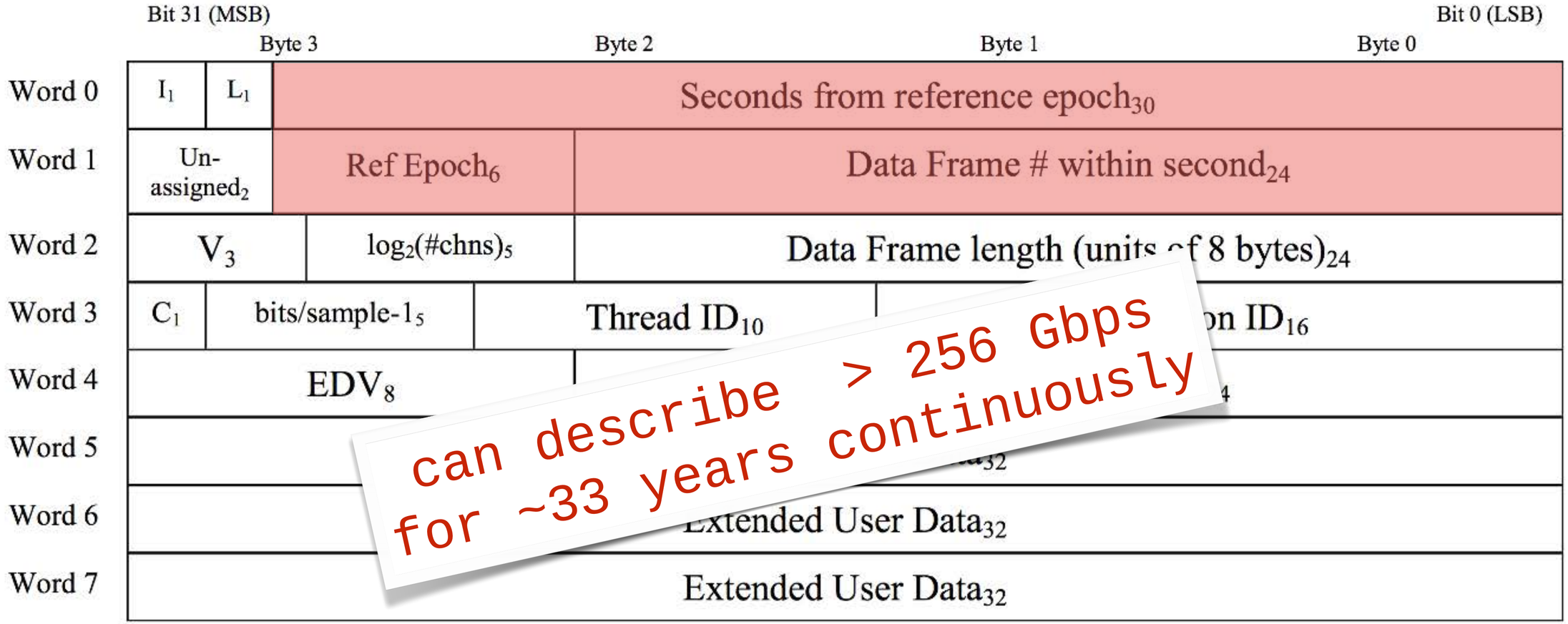


Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.



can describe > 256 Gbps for ~33 years continuously

Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

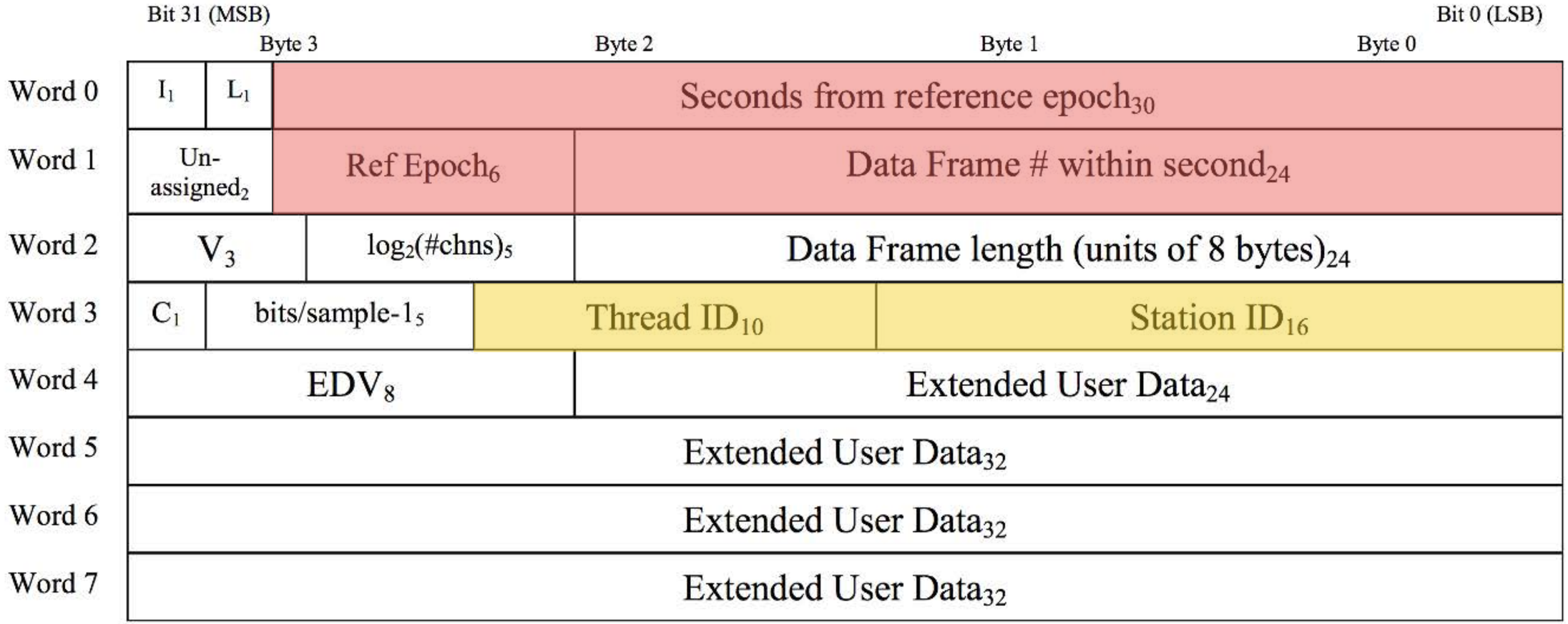


Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

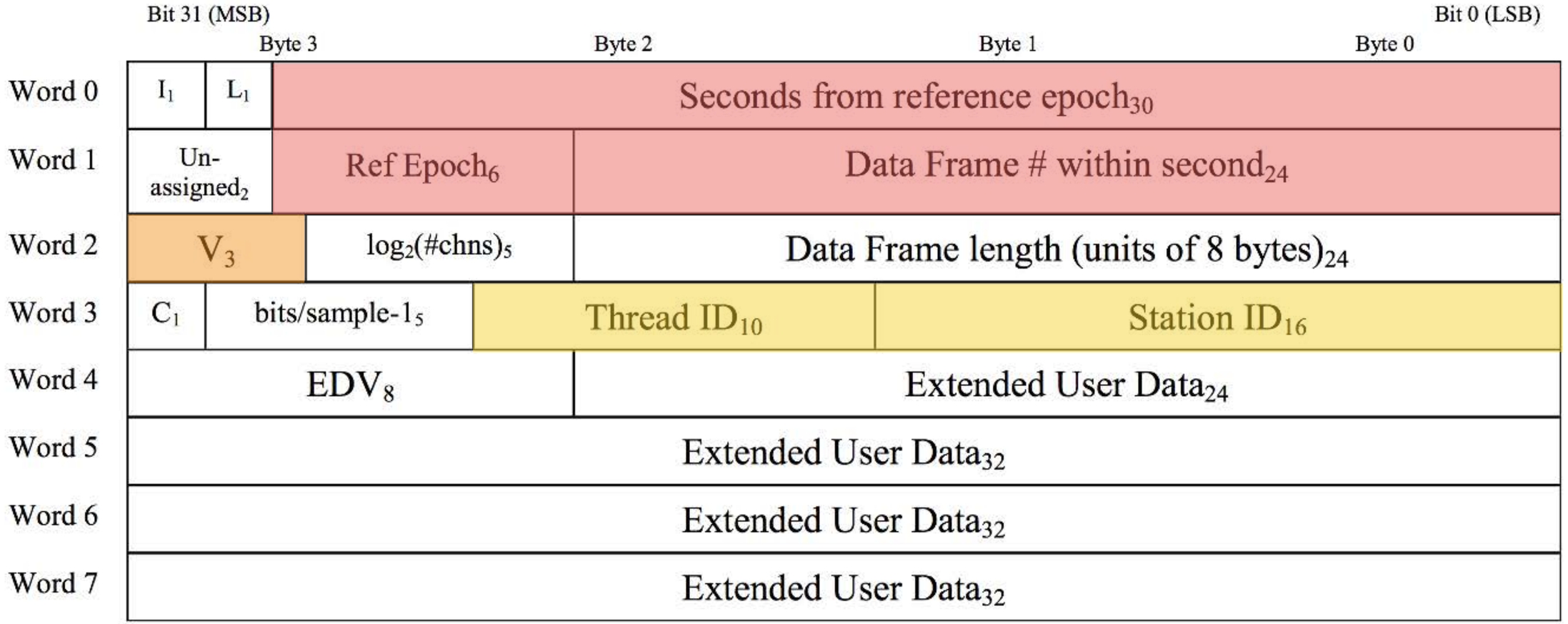


Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

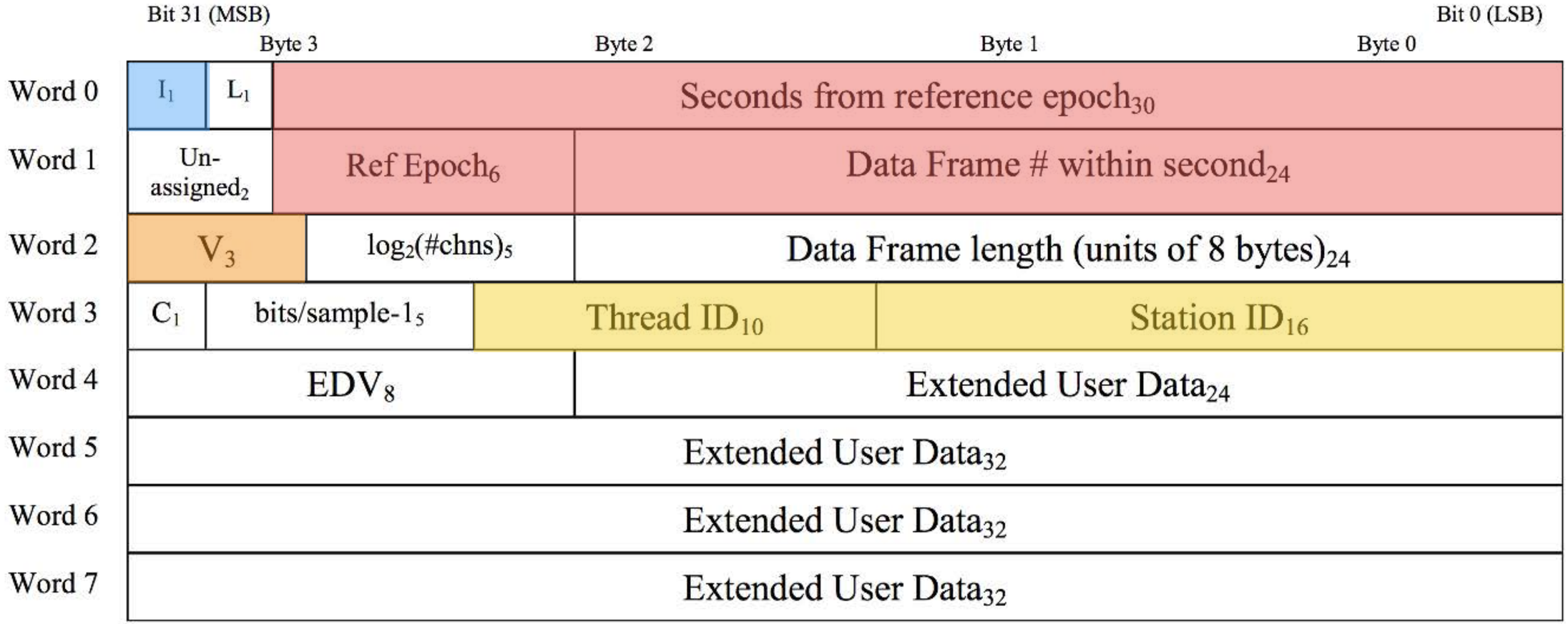


Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

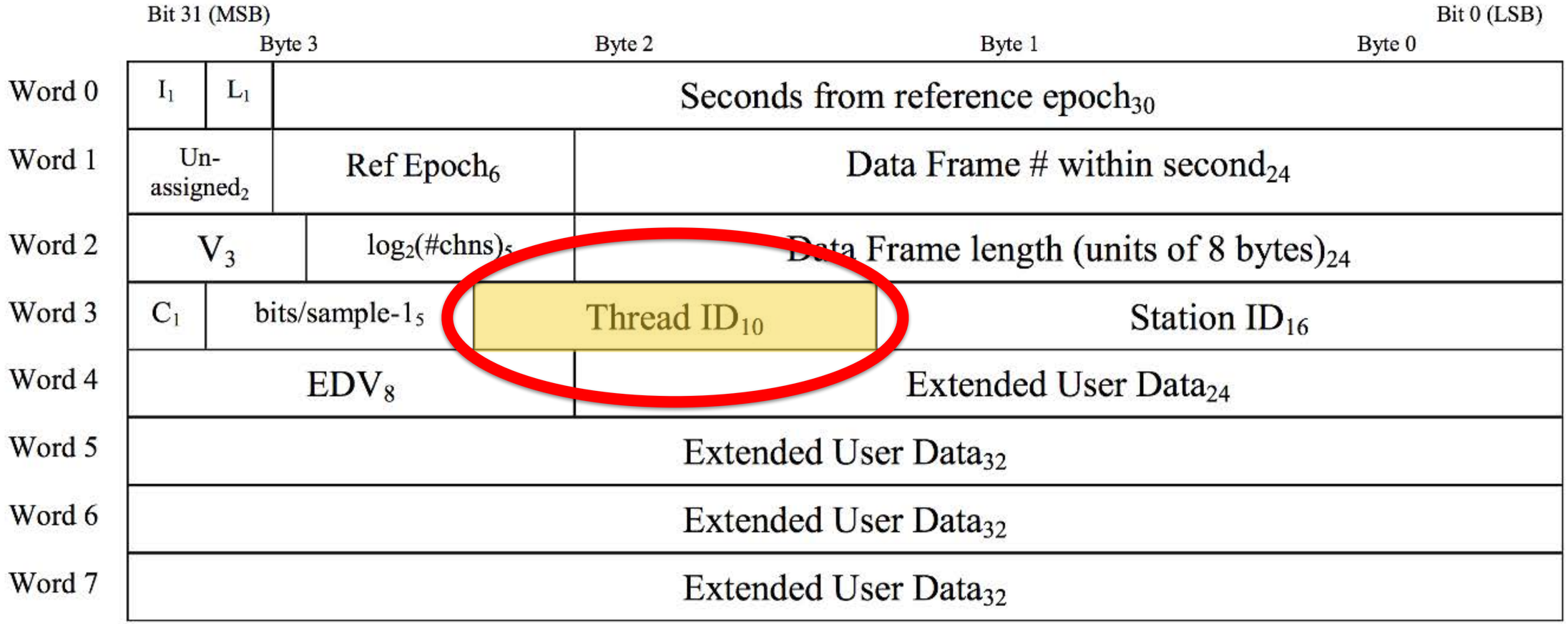
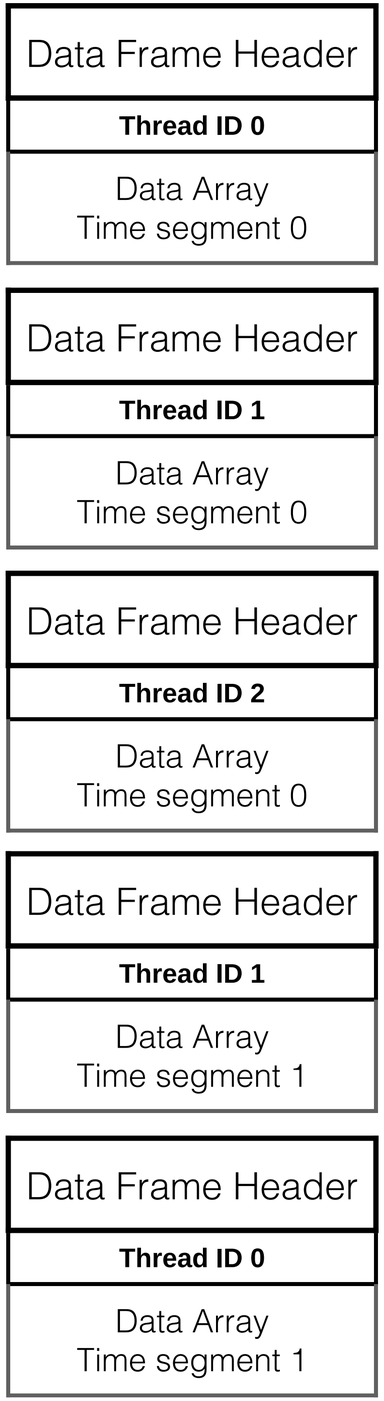
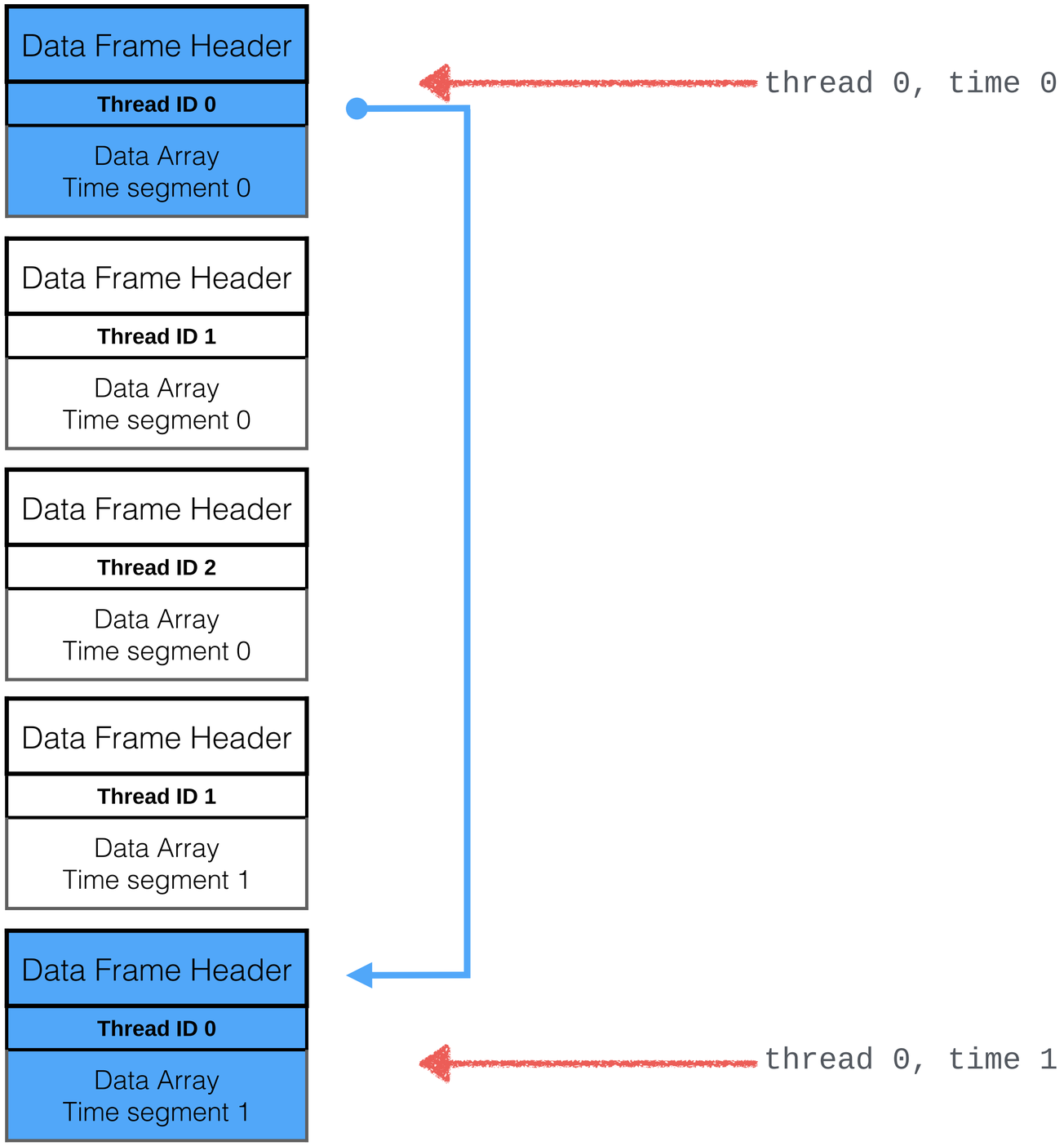
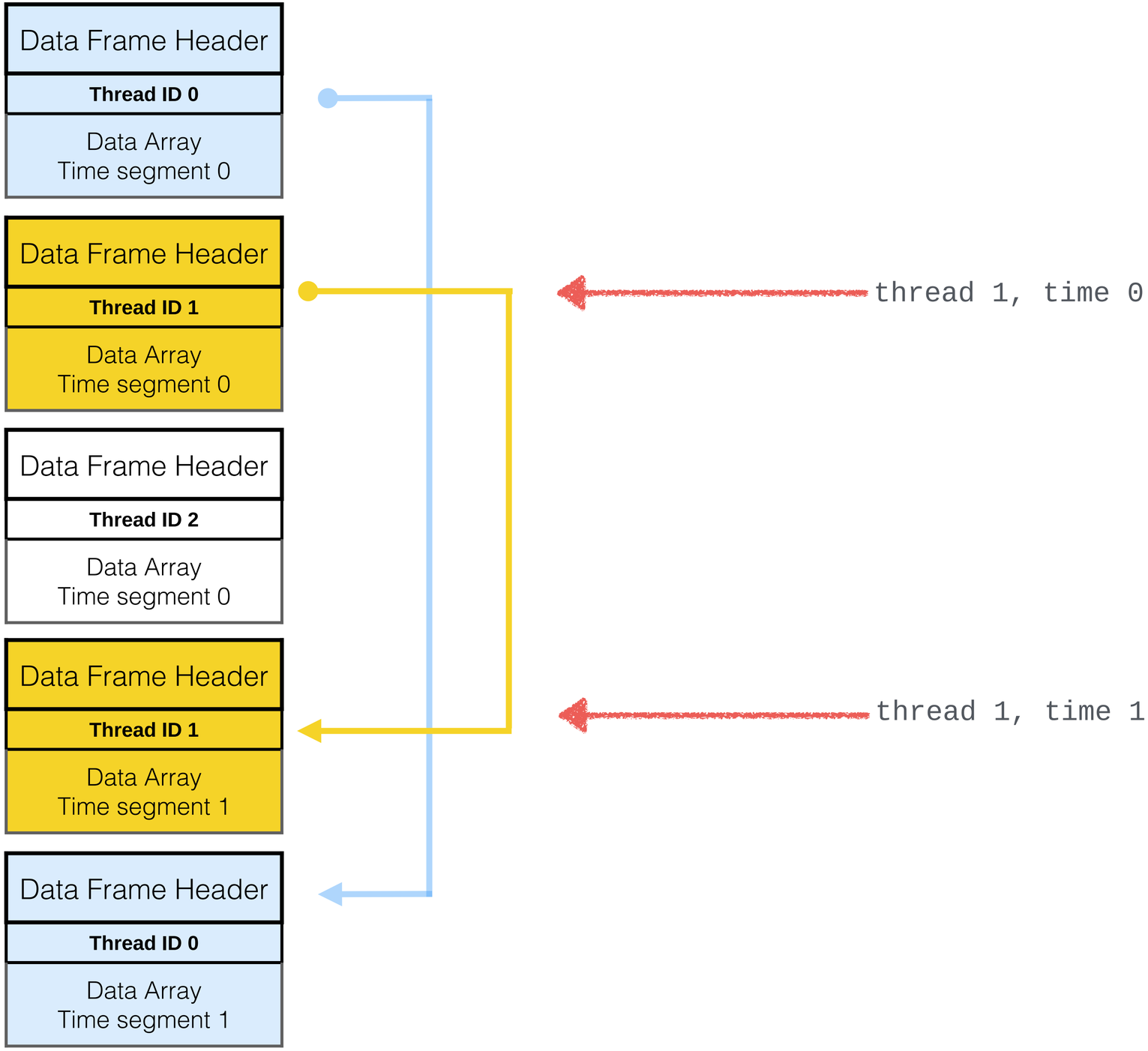


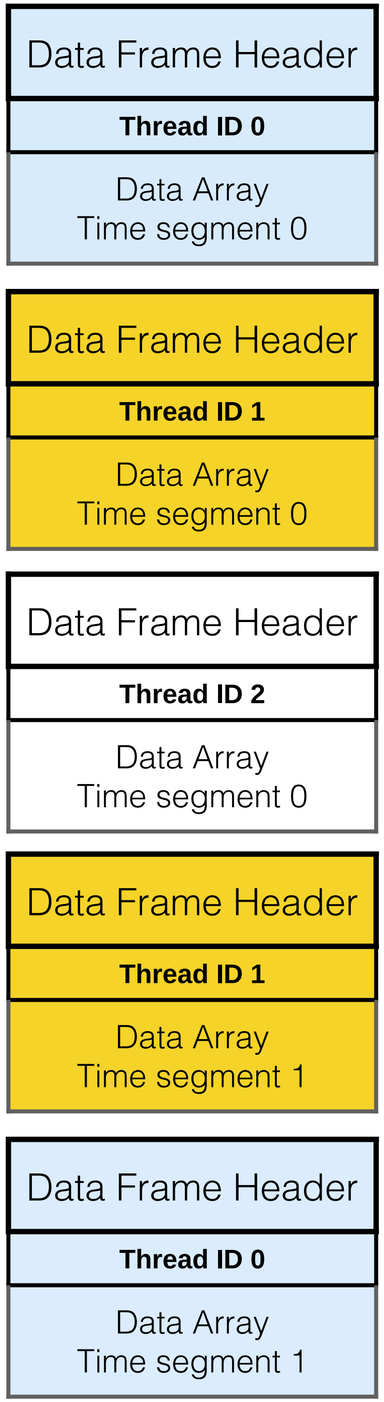
Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/


Data Frame Header
Thread ID 0
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 0
Data Frame Header
Thread ID 2
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 1
Data Frame Header
Thread ID 0
Data Array
Time segment 1


Data Frame Header
Thread ID 0
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 0
Data Frame Header
Thread ID 2
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 1
Data Frame Header
Thread ID 0
Data Array
Time segment 1
thread 0, time 0
thread 0, time 1


Data Frame Header
Thread ID 0
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 0
Data Frame Header
Thread ID 2
Data Array
Time segment 0
Data Frame Header
Thread ID 1
Data Array
Time segment 1
Data Frame Header
Thread ID 0
Data Array
Time segment 1
thread 1, time 0
thread 1, time 1



"The VDIF specification does not mandate strict Data Frame ordering within a Data Thread, but a best effort should be made to do so."

Data is recorded

…

now what?

Was the recording successful?

What the CENSORED did I just record?

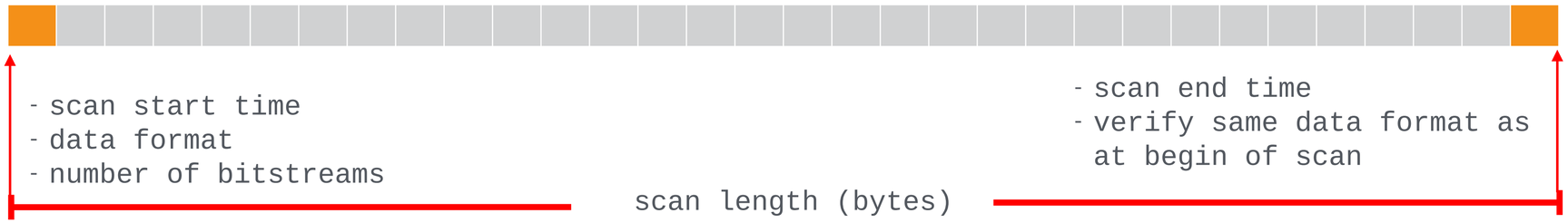
Enter: the **scan_check?** query

Was the recording successful?

What the CENSORED did I just record?

Enter: the **scan_check?** query

EXP_STA_SCAN(vbs/mk6)



Was the recording successful?

What the CENSORED did I just record?

Enter: the **scan_check?** query

- detect frame format (MarkIV, VLBA, Mark5B, VDIF)
  - mostly by looking for the magic bit pattern (syncword)
  - VDIF done heuristically so may #FAIL
- detect recorded data rate
- detect if data is missing
  - decode time stamp at begin and end of scan
  - *then* $((t_{end} - t_{begin}) \times data\_rate) - recording\_length$

Was the recording successful?

What the CENSORED did I just record?

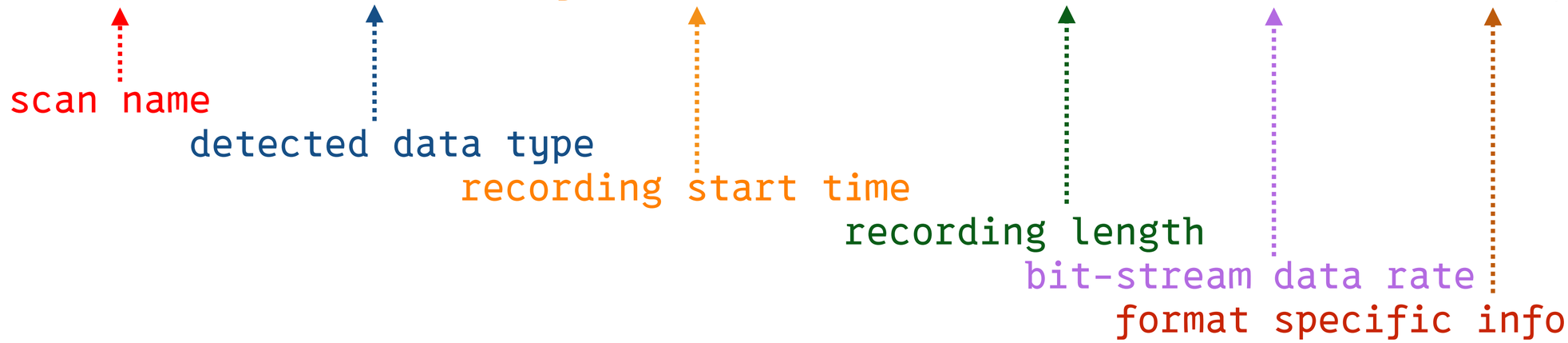
```
# Tell system to find a specific recording
# without argument (default) is ‘the last recording you made’
scan_set = n15x2_ef_no0014 ;
!scan_set = 0;


# What is in that recording?
scan_check?
!scan_check? 0 : ? : n15x2_ef_no0014 : VDIF : 2015y297d11h02m01.1389s : 780.786s : 16Mbps : 0 : 8032;
```

- scan name
- detected data type
- start time
- length
- data rate
- format specific information

Was the recording successful?

What the CENSORED did I just record?

```
# Tell system to find a specific recording
# without argument (default) is ‘the last recording you made’
scan_set = n15x2_ef_no0014 ;
!scan_set = 0;


# What is in that recording?
scan_check?
!scan_check? 0 : ? : n15x2_ef_no0014 : VDIF : 2015y297d11h02m01.1389s : 780.786s : 16Mbps : 0 : 8032;
```



Adaptable scan_check algorithm

The simplistic algorithm of “read at start and end” #FAIL

the **scan_check?** algorithm:

-re-invented + re-implemented

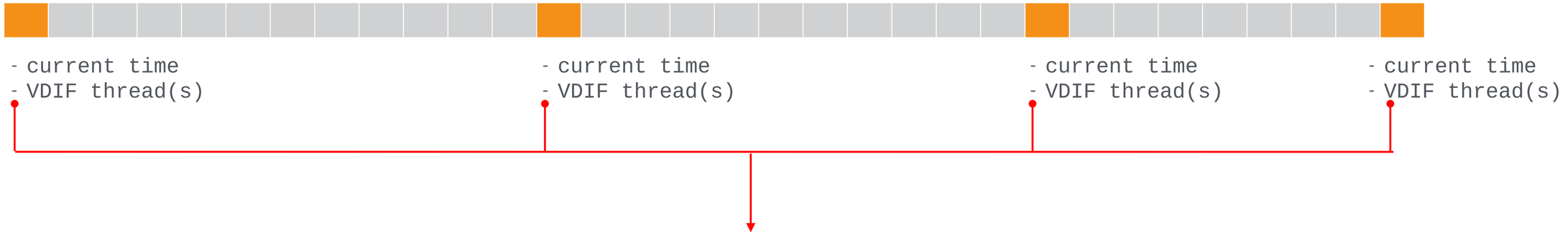
  -handle >1 VDIF threads in one “recording”

  -handle (very) high data rates

Adaptable scan_check algorithm

The simplistic algorithm of “read at start and end” #FAIL

EXP_STA_SCAN(vbs/mk6)



Combined result:

- VDIF threads
- data rate per thread

Adaptable scan_check algorithm

The simplistic algorithm of “read at start and end” #FAIL

EXP_STA_SCAN(vbs/mk6)

Allows tuning*

- how much data read in total
  - *more data read = better heuristics*, but:
  - decoding ~ 8MB takes ~ 1-2 s execution time!
- how many sample points/data per sampling point
- can take hint from current net_protocol setting
- will be in *jive5ab* 3.2 and up

(*) See e.g. https://github.com/jive-vlbi/jive5ab/commit/2304d5fb8e89463a8ea65e151a3eeec7180dbd23

Get data to correlator

What options do exist?

Get data to correlator

What options do exist? (Mark6, ship disk modules)

Get data to correlator

What options do exist?

```
$> scp /mnt/disk/* io13.mpifr-bonn.de:/path/…
```

???

Get data to correlator

What options do exist?

```
$> scp /mnt/disk/* io13.mpifr-bonn.de:/path/…
```

Get data to correlator

What options do exist?

```
$> ftp /mnt/disk/* io13.mpifr-bonn.de:/path/…
```

Get data to correlator

What options do exist?

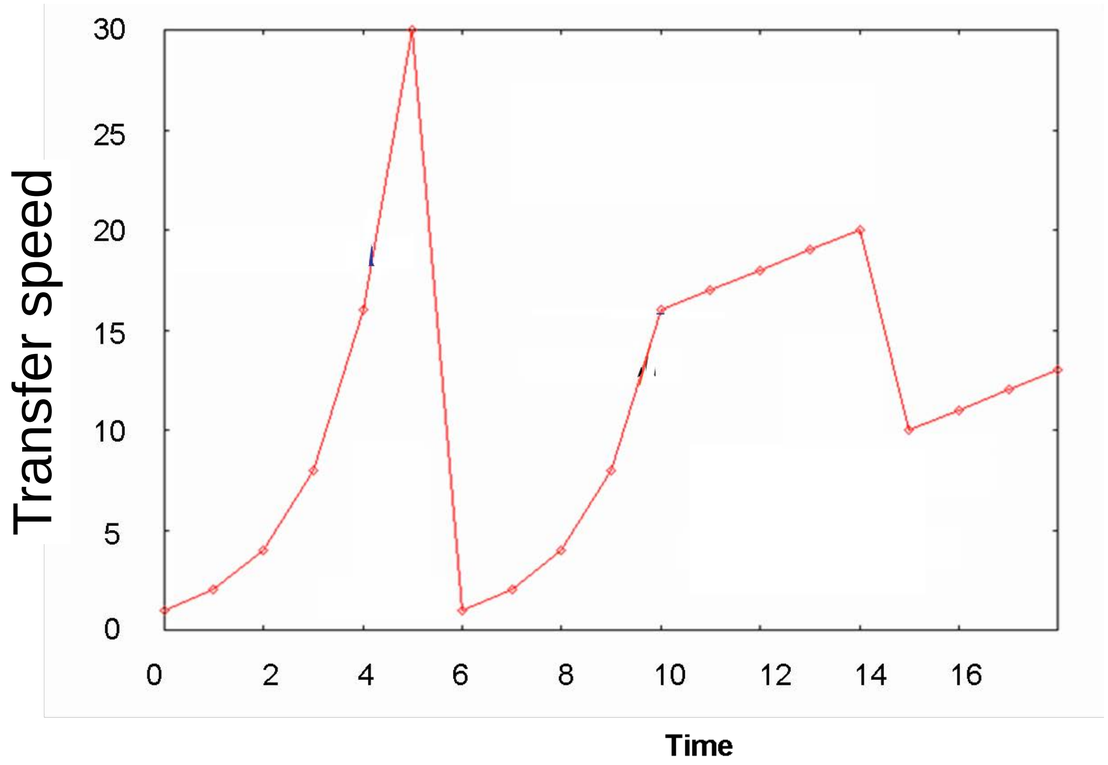
```
$> ftp|scp /mnt/disk/* io13.mpifr-bonn.de:/path/.
```

Transmission Control Protocol

https://en.wikipedia.org/wiki/Transmission_Control_Protocol

Get data to correlator

What options do exist?

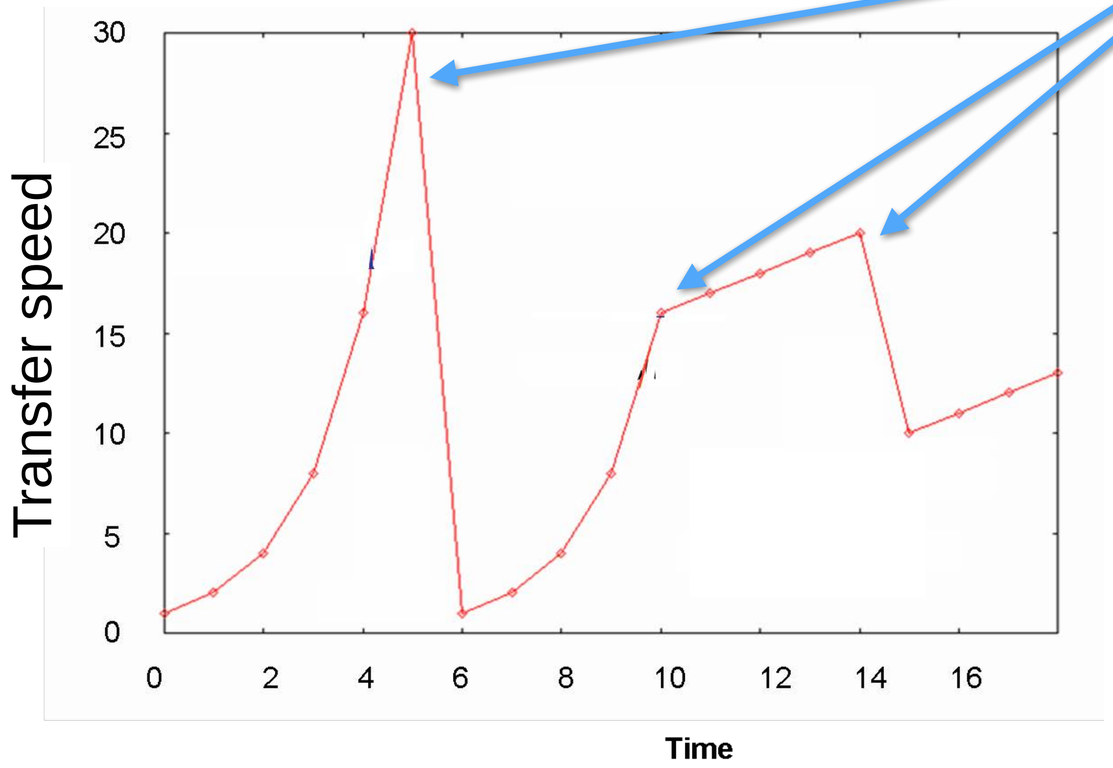


https://en.wikipedia.org/wiki/Transmission_Control_Protocol

Get data to correlator

What options do exist?

packet loss



https://en.wikipedia.org/wiki/Transmission_Control_Protocol

Get data to correlator

What options do exist? Use the **U**DP **D**ata **T**ransfer protocol

UDT

- software library in user space (not in O/S kernel!)
  - require application to actually *use* the library
- based on connectionless unreliable UDP protocol
- implement TCP-like features:
  - connection oriented
  - reliable

https://en.wikipedia.org/wiki/UDP-based_Data_Transfer_Protocol

Get data to correlator

What options do exist? Use the **U**DP **D**ata **T**ransfer protocol

UDT

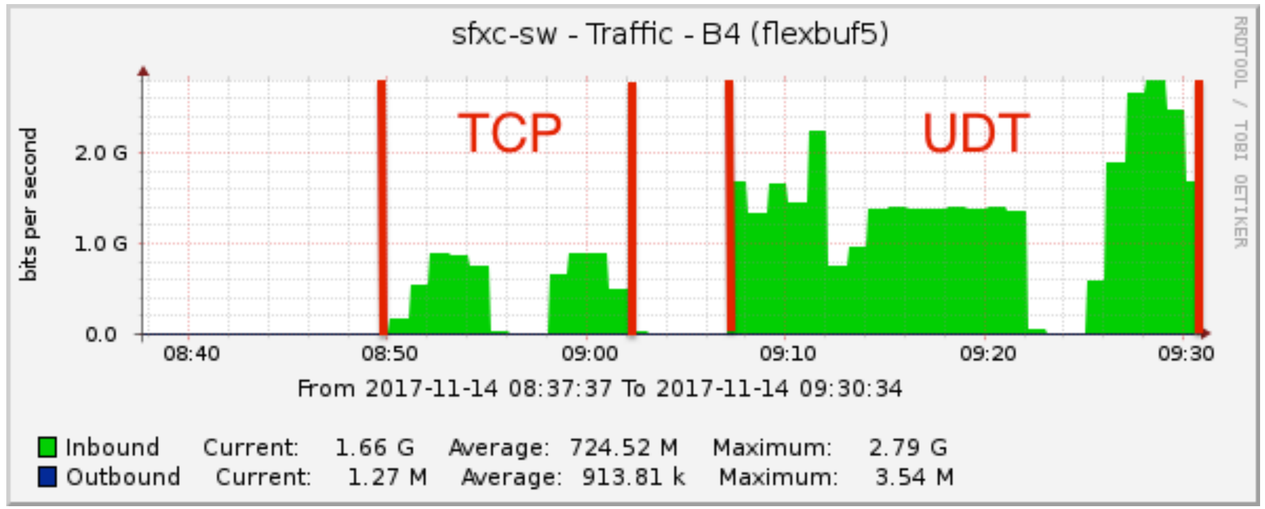
- software library in user space (not in O/S kernel!)
  - require application to actually *use* the library
- based on connectionless unreliable UDP protocol
- implement TCP-like features:
  - connection oriented
  - reliable
- A LOT faster on long fat links!

https://en.wikipedia.org/wiki/UDP-based_Data_Transfer_Protocol

Get data to correlator

What options do exist? Use the **U**DP **D**ata **T**ransfer protocol

UDT



https://en.wikipedia.org/wiki/UDP-based_Data_Transfer_Protocol

Get data to correlator

What options do exist? jive5ab has **UDT** support built in

jive5ab

net_protocol = **udt** : … ; (*)

(*) https://github.com/jive-vlbi/jive5ab/blob/master/doc/jive5ab-documentation-1.11.pdf

Get data to correlator

What options do exist? jive5ab e-shipping

https://github.com/jive-vlbi/jive5ab/blob/master/scripts/m5copy

Get data to correlator

What options do exist? jive5ab e-shipping

```
$> m5copy [options] SRC DST
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
$> m5copy [options] mk5:///1-10 file:///path/to/
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
$> m5copy [options] mk5:///1-10 file:///path/to/
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
$> m5copy [options] SRC DST
```

```
 mk5://[host][:port]/[module/]scans
file://[host][:port]/path/to/{dir/|file}
 mk6://[host][:port]/[module/]recording(s)
 vbs://[host][:port]/recording(s)
```

host: host name or IPv4 address (default **localhost**)
port: jive5ab command port (defaults **2620**)

Get data to correlator

What options do exist? jive5ab e-shipping

$> m5copy **[options]** SRC DST

[options]

`-p <PORT>`	PORT number to use for data connection (default **2630**)
`-m <MTU>`	MTU to use for the data transfer (default **1500**)
`-udt`	Use UDT as data transfer protocol (default **TCP**)
`-r <RATE>`	Maximum transfer rate to use (*only when using UDT*)

`--resume,`
`--allow_overwrite,`
`--ignore_existing`
	How to handle file(s)/recording(s) that already exist on the other side

Get data to correlator

What options do exist? jive5ab e-shipping

```
verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…
```

Get data to correlator

What options do exist? jive5ab e-shipping

```
verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…
```

Get data to correlator

What options do exist? jive5ab e-shipping

verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…

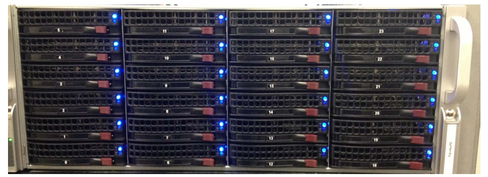
disk2net = connect : flexbuf12.jive.nl ;

tcp command channel, port 2621

tcp command channel, port 2620

net2vbs = open : … ;

udt data channel port 2631, mtu 9000



$> jive5ab -p 2621 –m 3 -6

$> jive5ab -m 3

Get data to correlator

What options do exist? jive5ab e-shipping **UDT firewall settings**

firewalls must allow
bi-directional UDP
traffic on **data port**

$> jive5ab -p 2621 –m 3 -6

udt data channel **port 2631**, mtu 9000

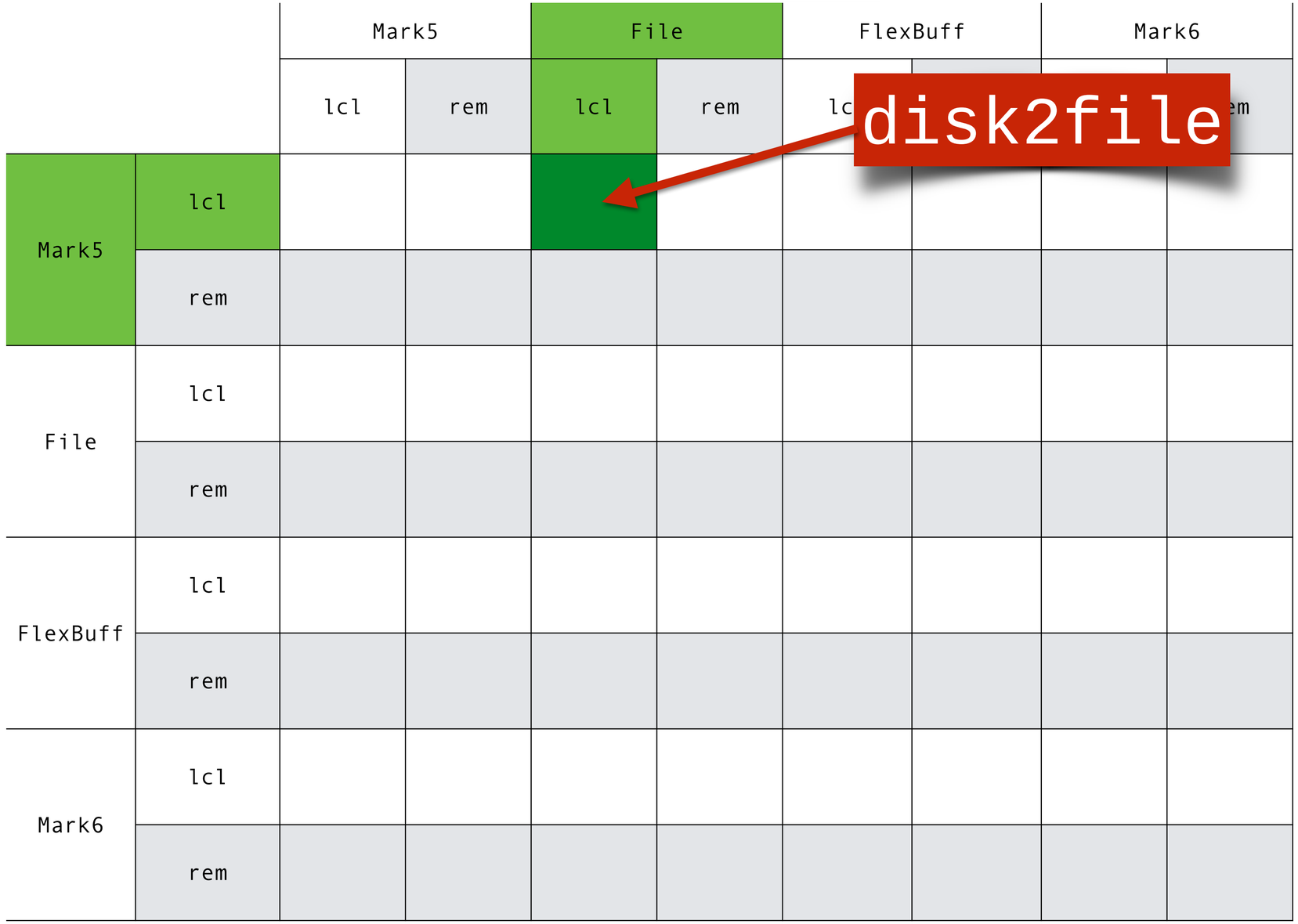
$> jive5ab -m 3

DST

SRC		Mark5		File		FlexBuff		Mark6	
		lcl	rem	lcl	rem	lcl	rem	lcl	rem
Mark5	lcl								
	rem								
File	lcl								
	rem								
FlexBuff	lcl								
	rem								
Mark6	lcl								
	rem								

DST

SRC


Mark5
File
FlexBuff
Mark6
lcl
rem
lcl
rem
lc
em
disk2file
Mark5
lcl
rem
File
lcl
rem
FlexBuff
lcl
rem
Mark6
lcl
rem

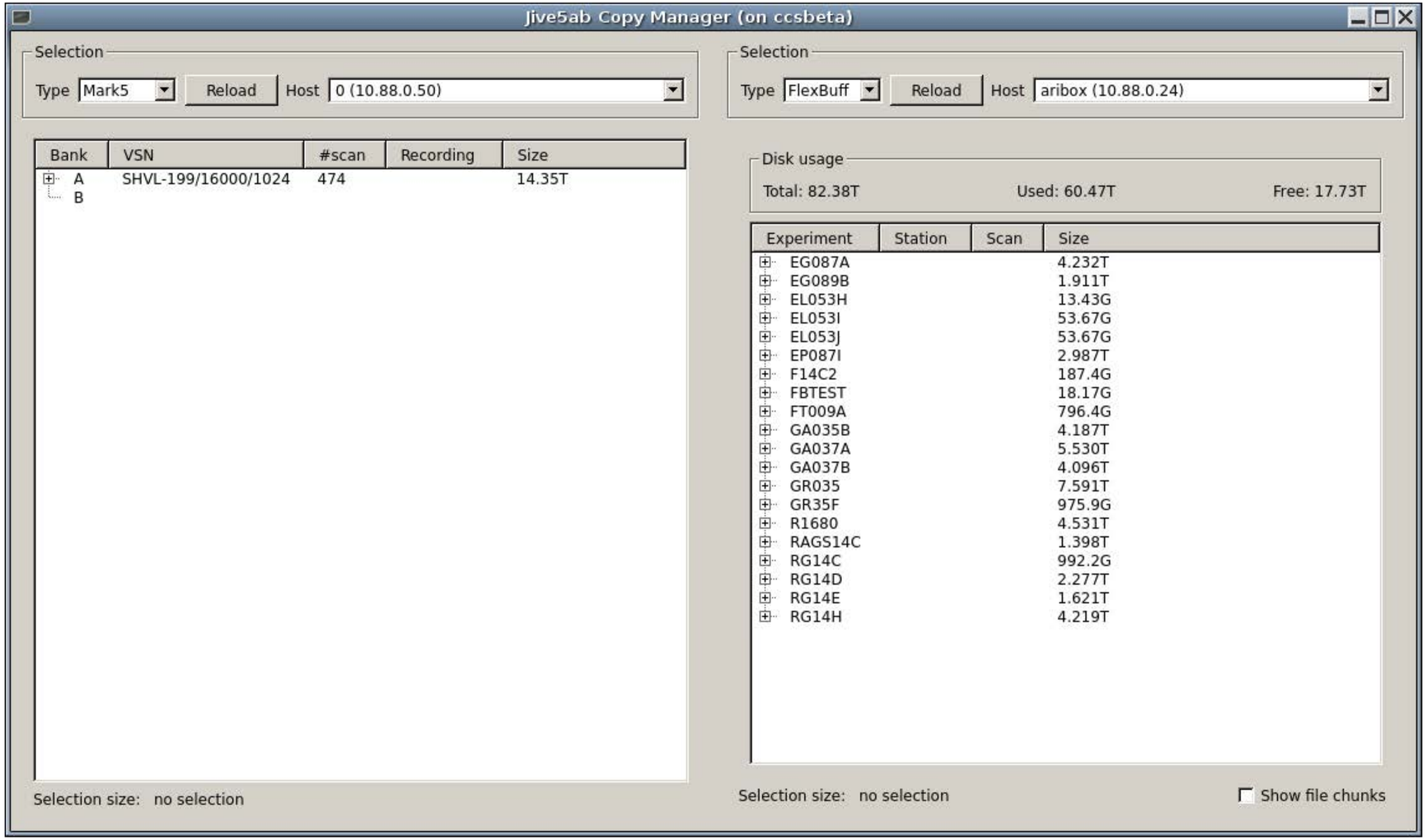
DST

SRC		Mark5		File		FlexBuff		Mark6	
		lcl	rem	lcl	rem	lcl	rem	lcl	rem
Mark5	lcl								
	rem								
File	lcl								
	rem								
FlexBuff	lcl								
	rem								
Mark6	lcl								
	rem								

disk2net + net2vbs

Get data to correlator

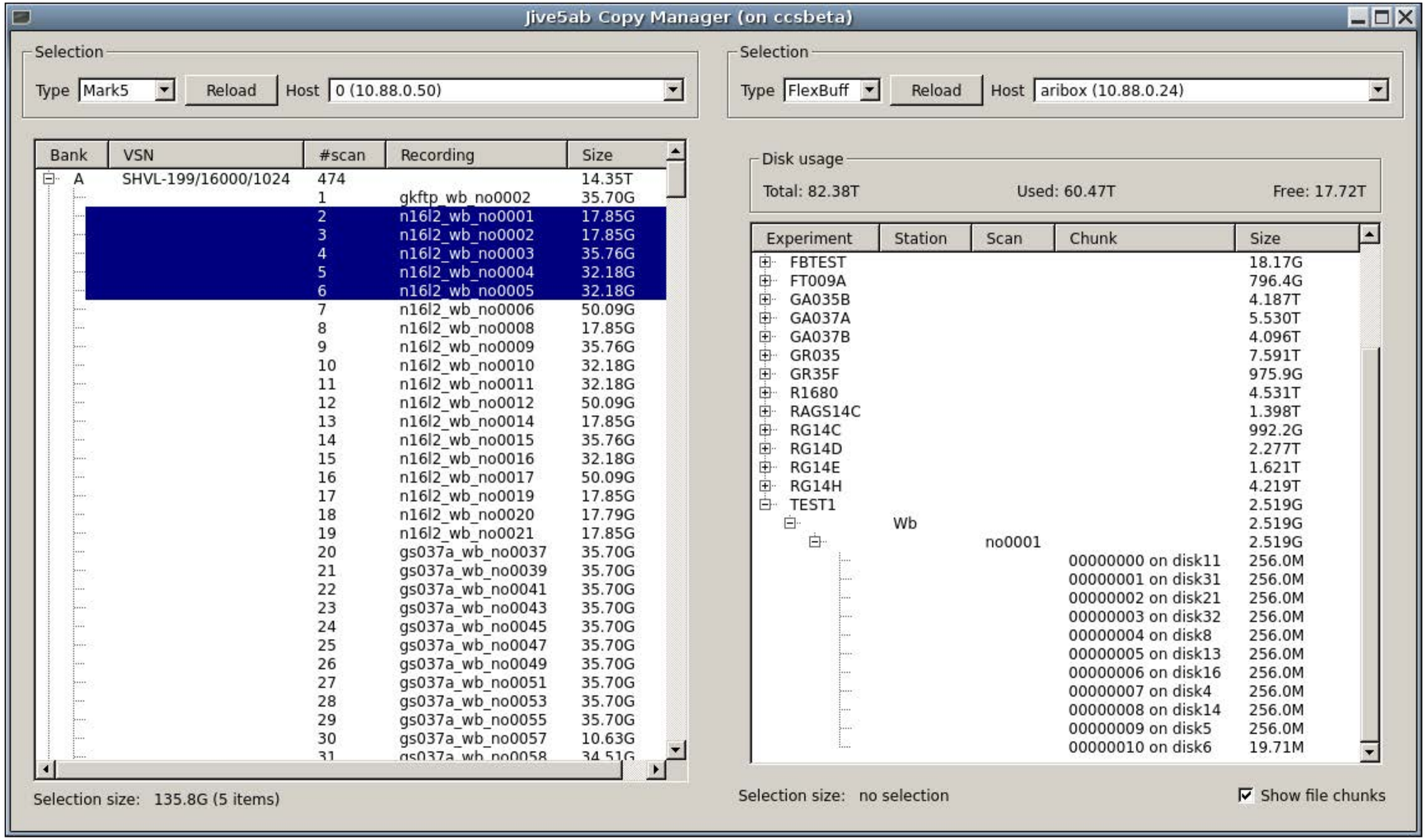
What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

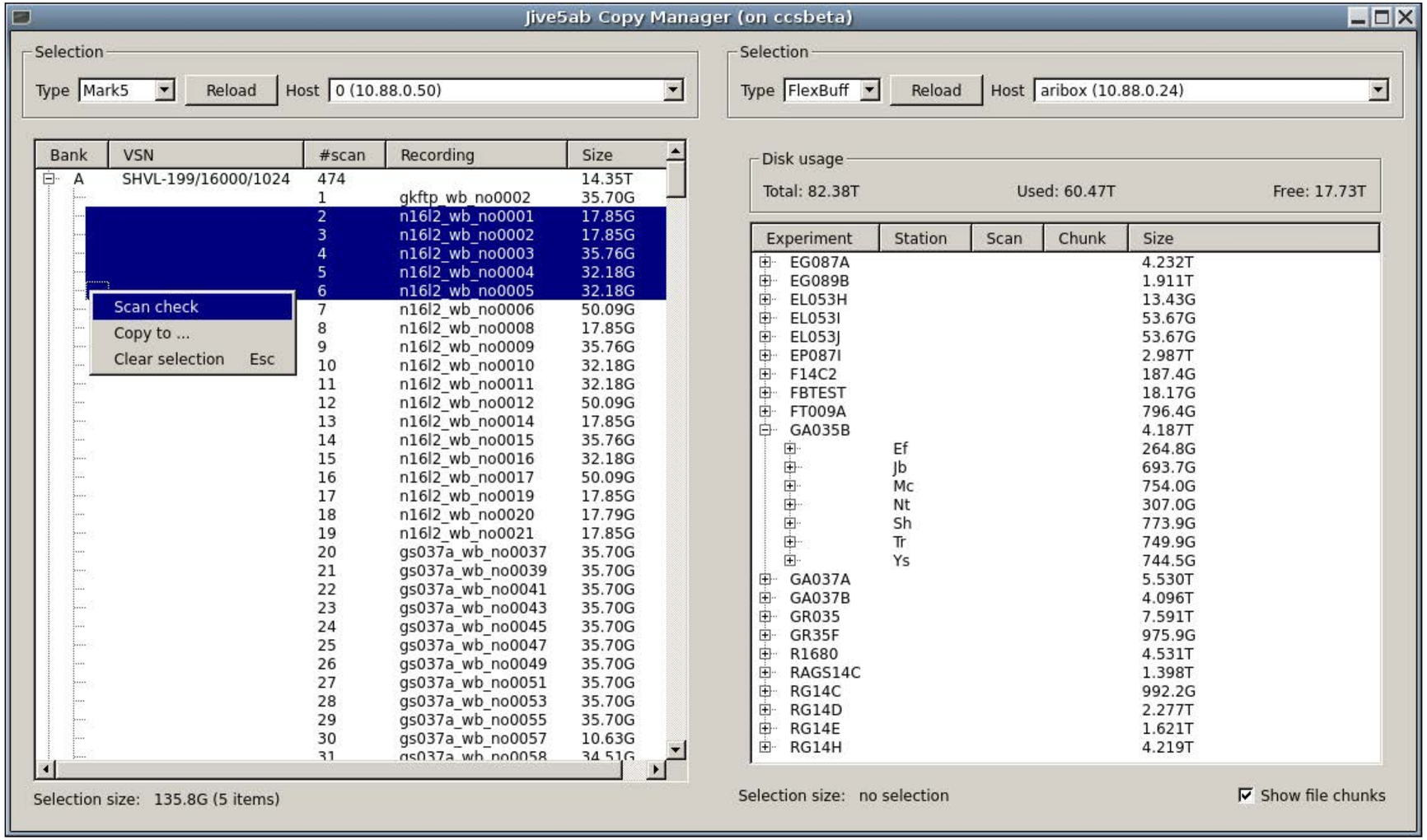
What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

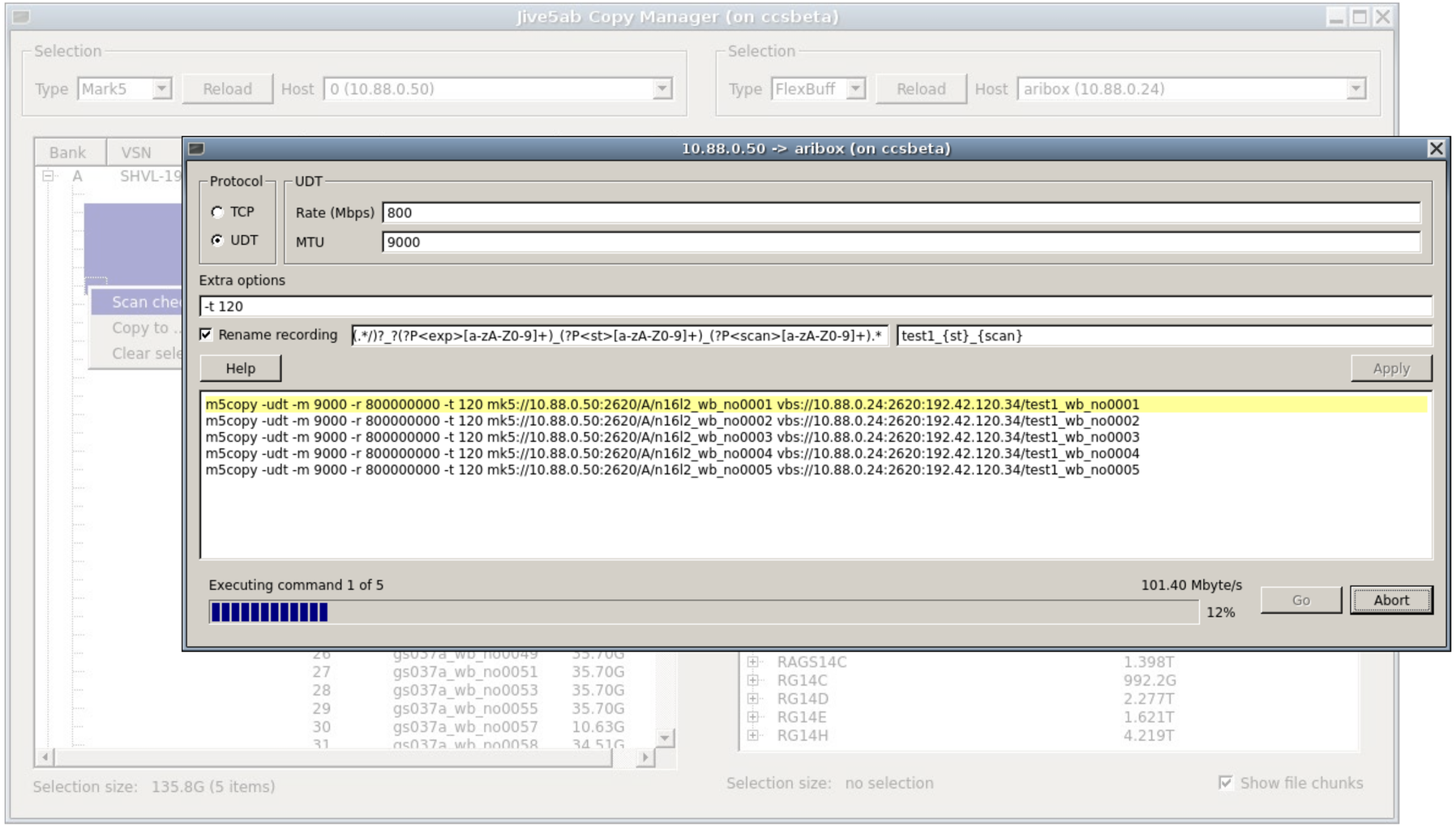
What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

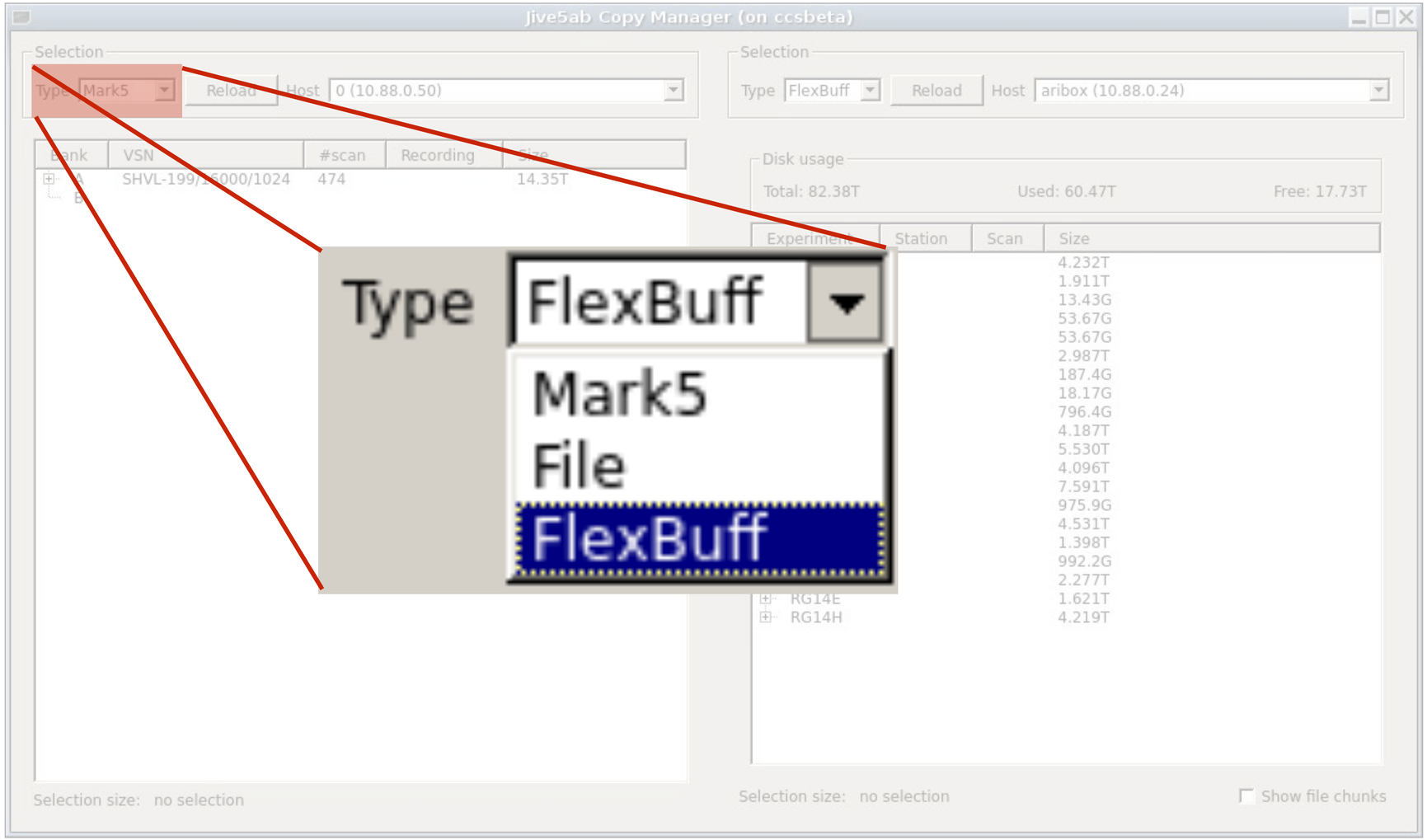
What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

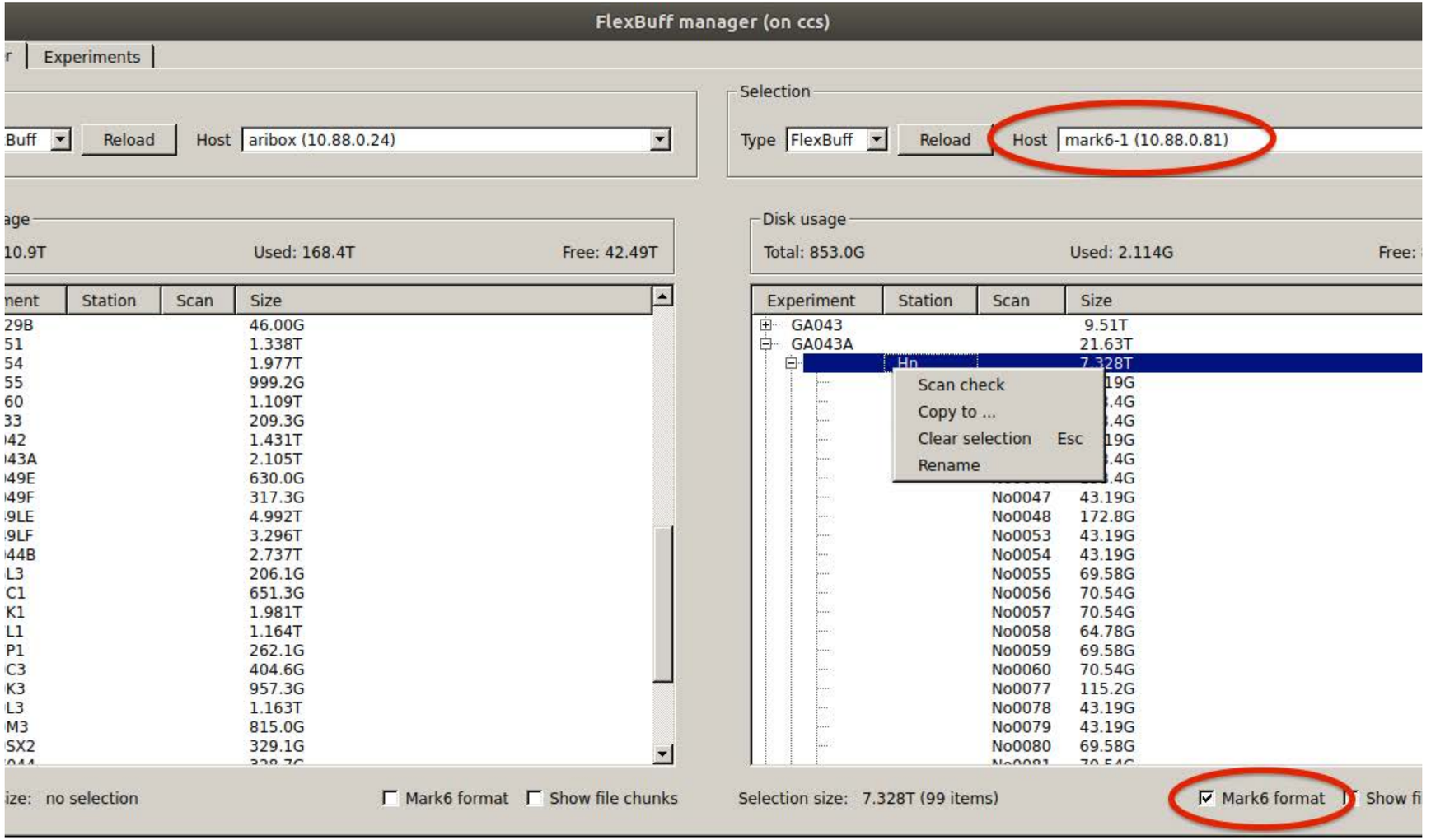
What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

What options do exist? jive5ab + m5copy **GUI frontend**



https://github.com/jive-vlbi/jive5ab-copy-manager

Get data to correlator

What options do exist? (jive5ab + m5copy e-shipping)

m5copy + 2 x jive5ab

Get data to correlator

The advantage of flexbuff-to-flexbuff e-shipping

Station

Correlator

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk3/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

/mnt/disk0/**EXP_STATION_SCAN/**

/mnt/disk1/**EXP_STATION_SCAN/**

Get data to correlator

The advantage of flexbuff-to-flexbuff e-shipping

Station

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk3/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

Correlator

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000** EXP_STATION_SCAN.**0000002**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

Get data to correlator

The advantage of flexbuff-to-flexbuff e-shipping

Station

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk3/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

Correlator

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000** EXP_STATION_SCAN.**0000002**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

Get data to correlator

The advantage of flexbuff-to-flexbuff e-shipping

Station

Correlator

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000006**

/mnt/disk3/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000005**

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000**

EXP_STATION_SCAN.**0000002**

EXP_STATION_SCAN.**0000007**

EXP_STATION_SCAN.**0000004**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001**

EXP_STATION_SCAN.**0000003**

EXP_STATION_SCAN.**0000006**

EXP_STATION_SCAN.**0000005**

Dealing with scattered data is a

Dealing with scattered data

Dealing with scattered data

How do I know what / how much is on those disks?!

Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, vbs_rm, vbs_fs

Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, vbs_rm, vbs_fs

Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, **vbs_rm**, vbs_fs

Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, vbs_rm, **vbs_fs**

Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, **vbs_rm**, vbs_fs

Python scripts, list/remove vbs+mk6
modelled after **ls(1)** and **rm(1)**
many familiar options

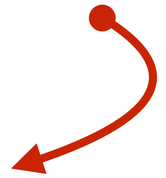
Dealing with scattered data

How do I know what / how much is on those disks?!

VBS_FS tools

https://code.jive.eu/verkout/vbs_fs

vbs_ls, vbs_rm, **vbs_fs**



FUSE virtual file system driver

https://en.wikipedia.org/wiki/Filesystem_in_Userspace

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$>
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls [options][pattern…]
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls -lrth haavee*
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls -lrth haavee*
Found 4 recordings in 90 chunks,   57.68G
drw-r--r-- jops      flexbuf     12.75G Jun 22 10:27 haavee_vbs_no0001
drw-r--r-- jops      flexbuf      9.75G Jun 22 10:27 haavee_vbs_no0001a
-rw-r--r-- jops      flexbuf     15.64G Jun 22 10:27 haavee_m6_no0001
-rw-r--r-- jops      flexbuf     19.54G Jun 22 10:28 haavee_m6_no0002
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls -lrth haavee*
Found 4 recordings in 90 chunks,   57.68G
drw-r--r-- jops      flexbuf     12.75G Jun 22 10:27 haavee_vbs_no0001
drw-r--r-- jops      flexbuf      9.75G Jun 22 10:27 haavee_vbs_no0001a
-rw-r--r-- jops      flexbuf     15.64G Jun 22 10:27 haavee_m6_no0001
-rw-r--r-- jops      flexbuf     19.54G Jun 22 10:28 haavee_m6_no0002
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls -lrth haavee*
Found 4 recordings in 90 chunks,   57.68G
drw-r--r-- jops      flexbuf     12.75G Jun 22 10:27 haavee_vbs_no0001
drw-r--r-- jops      flexbuf      9.75G Jun 22 10:27 haavee_vbs_no0001a
-rw-r--r-- jops      flexbuf     15.64G Jun 22 10:27 haavee_m6_no0001
-rw-r--r-- jops      flexbuf     19.54G Jun 22 10:28 haavee_m6_no0002
```

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls …  -T …  <pattern> [<pattern> …]
```

“accumulate by <pattern>s”

Dealing with scattered data

How do I know what / how much is on those disks?!

```
$> vbs_ls … -lTh em117e* em117f* eg088_ys*
```

“accumulate by <pattern>s”

```
Found 3 recordings in 13426 chunks,    3.11T
drw-r--r-- jops      flexbuf      2.36T Aug 05 13:24 eg088_ys*
drw-r--r-- jops      flexbuf    394.59G Jun 06 20:34 em117e*
drw-r--r-- jops      flexbuf    372.59G Jun 06 18:09 em117f*
```

Dealing with scattered data

Easy access to recordings using a virtual file system

```
verkout@flexbuf1:$ find . -type f -name haavee\* -exec ls -lh {} \;
```

Dealing with scattered data

Easy access to recordings using a virtual file system

```
verkout@flexbuf1:$ find . -type f -name haavee\* -exec ls -lh {} \;

-rw-r--r-- 1 jops flexbuf 3.3G Jun 22 12:28 ./haavee_m6_no0002
-rw-r--r-- 1 jops flexbuf 2.7G Jun 22 12:27 ./haavee_m6_no0001

-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:26 ./haavee_vbs_no0001/haavee_vbs_no0001.00000007
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:26 ./haavee_vbs_no0001/haavee_vbs_no0001.00000019
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:26 ./haavee_vbs_no0001/haavee_vbs_no0001.00000013
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:26 ./haavee_vbs_no0001/haavee_vbs_no0001.00000025
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001/haavee_vbs_no0001.00000031
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:26 ./haavee_vbs_no0001/haavee_vbs_no0001.00000001
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001/haavee_vbs_no0001.00000049
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001/haavee_vbs_no0001.00000043
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001/haavee_vbs_no0001.00000037

-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001a/haavee_vbs_no0001a.00000037
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001a/haavee_vbs_no0001a.00000001
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001a/haavee_vbs_no0001a.00000031
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001a/haavee_vbs_no0001a.00000025
-rw-r--r-- 1 jops flexbuf 256M Jun 22 12:27 ./haavee_vbs_no0001a/haavee_vbs_no0001a.00000019
```

Dealing with scattered data

Easy access to recordings using a virtual file system

```
$> vbs_fs [options] /path/to/mountpoint
```

Dealing with scattered data

Easy access to recordings using a virtual file system

/mnt/disk/1/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/2/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/3/0/data/**EXP_STATION_SCAN.vdif**

/mnt/disk/4/7/data/**EXP_STATION_SCAN.vdif**

/mnt/disk0/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000000** EXP_STATION_SCAN.**0000007**

/mnt/disk1/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000002** EXP_STATION_SCAN.**0000004**

/mnt/disk2/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000001** EXP_STATION_SCAN.**0000006**

/mnt/disk3/**EXP_STATION_SCAN/**

EXP_STATION_SCAN.**0000003** EXP_STATION_SCAN.**0000005**

/path/to/mountpoint/**EXP_STATION_SCAN.vdif**

DATA**[0]** DATA**[1]** DATA**[2]** DATA**[3]** DATA**[4]** DATA**[5]** DATA**[6]** **...**

/path/to/mountpoint/**EXP_STATION_SCAN**

EXP_STATION_SCAN.**0000000** EXP_STATION_SCAN.**0000001** EXP_STATION_SCAN.**0000002** EXP_STATION_SCAN.**0000003** **...**

Dealing with scattered data

Easy access to recordings using a virtual file system

```
$> vbs_fs [options] /path/to/mountpoint
```

A multi-threaded FUSE virtual file system in C++

- Reconstructs scattered recordings as single files, transparently for:
  - `cplane/dplane` MIT Haystack Mark6 format
  - `jive5ab` FlexBuff/vbs format
- User, group, permissions, modification time reflect actual status of on-disk files
- I/O scheduling done in vbs_fs
  - configurable read-ahead
  - serving multiple files and/or multiple users guaranteed optimal *if reading from the same mountpoint*
  - `vbs_fs` can be run multiple times but may degrade I/O performance depending on usage pattern

Dealing with scattered data

Easy access to recordings using a virtual file system

```
$> mkdir /path/to/mountpoint
```

```
$> vbs_fs [options] /path/to/mountpoint
```

```
$> ls -alh /path/to/mountpoint
-rw-r--r-- 0 jops flexbuf  16G Jun 22 12:27 /tmp/foo/haavee_m6_no0001
-rw-r--r-- 0 jops flexbuf  20G Jun 22 12:28 /tmp/foo/haavee_m6_no0002
-rw-r--r-- 0 jops flexbuf  13G Jun 22 12:27 /tmp/foo/haavee_vbs_no0001
-rw-r--r-- 0 jops flexbuf 9.8G Jun 22 12:27 /tmp/foo/haavee_vbs_no0001a
```

Dealing with scattered data

Easy access to recordings using a virtual file system

$> mkdir /path/to/mountpoint

$> vbs_fs /path/to/mountpoint

$> ls -alh /path/to/mountpoint

100% VDIF useful payload only!

vbs_fs strips all cplane headers

```
-rw-r--r-- 0 jops flexbuf  16G Jun 22 12:27 /tmp/foo/haavee_m6_no0001
-rw-r--r-- 0 jops flexbuf  20G Jun 22 12:28 /tmp/foo/haavee_m6_no0002
-rw-r--r-- 0 jops flexbuf  13G Jun 22 12:27 /tmp/foo/haavee_vbs_no0001
-rw-r--r-- 0 jops flexbuf 9.8G Jun 22 12:27 /tmp/foo/haavee_vbs_no0001a
```

Dealing with scattered data

Easy access to recordings using a virtual file system

```
$> mkdir /path/to/mountpoint
```

```
$> vbs_fs [options] /path/to/mountpoint
```

[options]

`-6`	Scan Mark6 mountpoints for *shrapnel* (default **FlexBuff mountpoints**)
`-R <PATH>`	Add `<PATH>` to list of directories to scan for *shrapnel*
`-I <PATTERN>`	Only index/scan recordings matching `<PATTERN>` (default **anything recognizeable as recording**)
`-n <NUM>`	Enable read-ahead of `<NUM>` blocks to speed up data access (default **no read-ahead**)

Get data to correlator

What options do exist? e-transfer daemon/client

Get data to correlator

What options do exist? e-transfer daemon/client

```
verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…
```

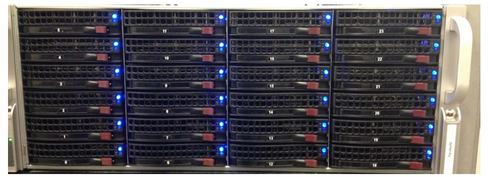
```
disk2net = connect : flexbuf12.jive.nl ;
```

tcp command channel, port 2621

tcp command channel, port 2620

```
net2vbs = open : … ;
```

udt data channel port 2631, mtu 9000



```
$> jive5ab -p 2621 –m 3 -6
```

```
$> jive5ab -m 3
```

Get data to correlator

What options do exist? e-transfer daemon/client

verkout@Mac> m5copy -udt -mtu 9000 -p 2631 mk6://130.141.242.16:2621/… vbs://flexbuf12.jive.nl/…

disk2net = connect : flexbuf12.jive.nl ;

net2vbs = open : … ;

tcp channel, port 2621

command

tcp channel, port 2620

command

udt **data** channel port 2631, mtu 9000

$> jive5ab -p 2621 –m 3 -6

$> jive5ab -m 3

Get data to correlator

What options do exist? e-transfer daemon/client

The e-transfer tools

https://github.com/jive-vlbi/etransfer

etd, etc

“e-transfer daemon”
“e-transfer client”

Get data to correlator

What options do exist? e-transfer daemon/client

The e-transfer tools support

- tcp, udt
  - IPv4 + IPv6
- proper daemon
  - multiple clients over one port simultaneously
  - server communicates data channel to client
- also remote-to-remote transfers
- file to file only

Get data to correlator

What options do exist? e-transfer daemon/client

The e-transfer tools sup

- tcp, udt
  - IPv4 + IP
- p
- m
- file to file only

in m5copy the user sets the data channel, not the server admin

port simultaneously
data channel to client

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etd [options] --command tcp:// --data udt://
```

Get data to correlator

What options do exist? e-transfer daemon/client

$> …/etd [options] **--command tcp://** --data udt://

--command <protocol>://[<host>][:<port>]

<protocol>	tcp, tcp6, udt, udt6 (default **no default**)
<host>	IPv4 or IPv6 address or hostname (default **all interfaces** i.e. **IPv4 0.0.0.0**)
<port>	Port number to listen on for incoming command connections (default **4004**)

Get data to correlator

What options do exist? e-transfer daemon/client

$> …/etd [options] --command tcp:// **--data udt://**

--data <protocol>://[<host>][:<port>]

<protocol>	tcp, tcp6, udt, udt6 (default **no default**)
<host>	IPv4 or IPv6 address or hostname (default **all interfaces** i.e. **IPv4 0.0.0.0**)
<port>	Port number to listen on for incoming data connections (default **8008**)

Get data to correlator

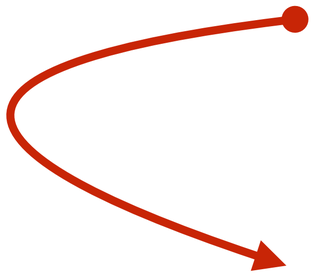
What options do exist? e-transfer daemon/client

```
$> …/etd [options] --command tcp://
        --data tcp://10.88.0.33:8009
        --data udt://192.42.120.39:8008
```

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etd [options] --command tcp://
        --data tcp://10.88.0.33:8009
        --data udt://192.42.120.39:8008
```



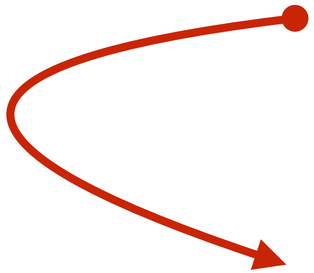
e-transfer client will:

- try to connect to data channels in this order

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etd [options] --command tcp://
        --data tcp://10.88.0.33:8009
        --data udt://192.42.120.39:8008
```



e-transfer client will:

- try to connect to data channels in this order
- internal client uses fast tcp

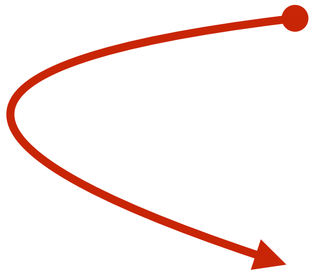
Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etd [options] --command tcp://
```

~~**--data tcp://10.88.0.33:8009**~~

--data udt://192.42.120.39:8008



e-transfer client will:

- try to connect to data channels in this order
- internal client uses fast tcp
- external client uses fast udt

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etc [options] SRC DST
```

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etc [options] SRC DST
```

```
/path/to/file*.*
```

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etc [options] SRC DST
```

```
/path/to/file*.*
flexbuf4.jive.nl:/path/to/file*.*
flexbuf4.jive.nl#4005:/path/to/file*.*
tcp6:flexbuf6.jive.nl:/path/to/file*.*
```

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> …/etc [options] SRC DST

  /path/to/file*.*
  flexbuf4.jive.nl:/path/to/file*.*
  flexbuf4.jive.nl#4005:/path/to/file*.*
  tcp6:flexbuf6.jive.nl:/path/to/file*.*
```

non-standard control port

non-standard protocol

Get data to correlator

What options do exist? e-transfer daemon/client

$> …/etc [options] SRC **DST**

/path/to/{dir/|file}
130.141.242.16:/path/to/{dir/|file}
udt:fb7.jive.nl**#42267:**/path/to/{dir/|file}

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> vbs_fs [options] /mnt/data
```

Get data to correlator

What options do exist? e-transfer daemon/client

```
$> vbs_fs [options] /mnt/data
$> …/etc [options] /mnt/data/* fb7.jive.nl:/data/
```

Get data to correlator

What options do exist? e-transfer daemon/client

$> …/etc **[options]** SRC DST

[options]

`-h, --help`	short (long) explanation of the command line
`-m <number>`	Message level - higher number = more output (default **0**)
`-v`	Enable verbose output on each file transferred

`--resume,`
`--overwrite,`
`--skipexisting`
	How to handle file(s) that already exist on the other side (default **New** i.e. *file may not exist*!)

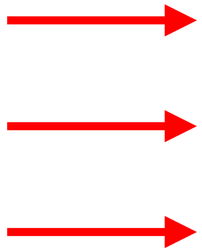
… and there's more …

Thanks for attention!

(Automatically) collect frames in multiple recordings?

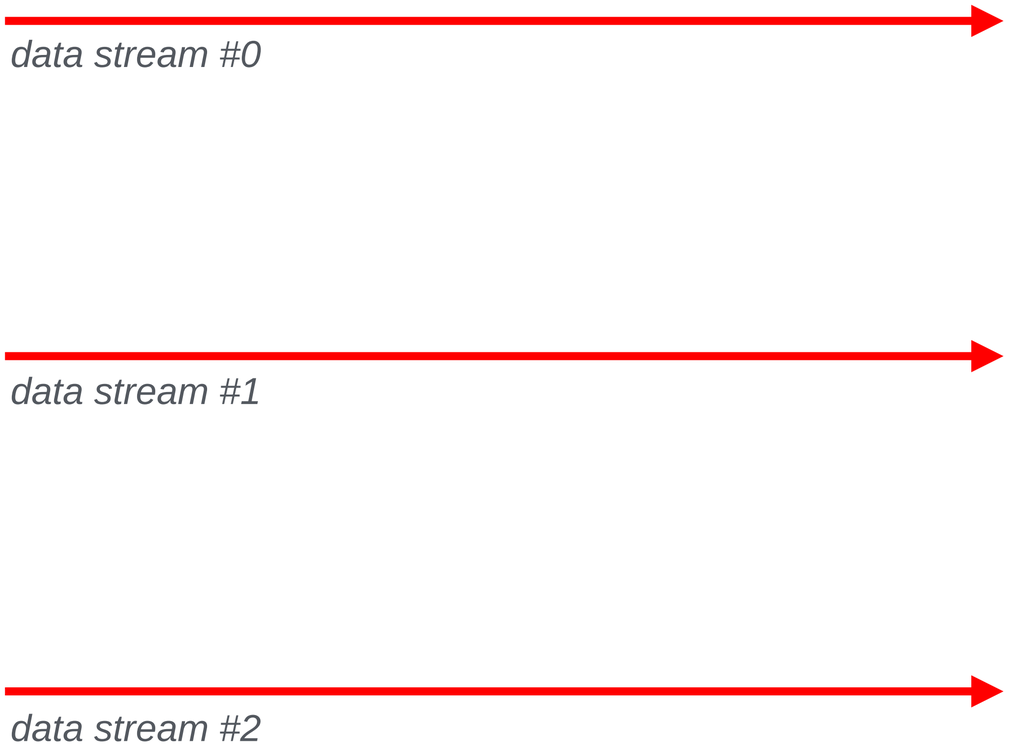
Extra Good Stuff™

digital backend(s)



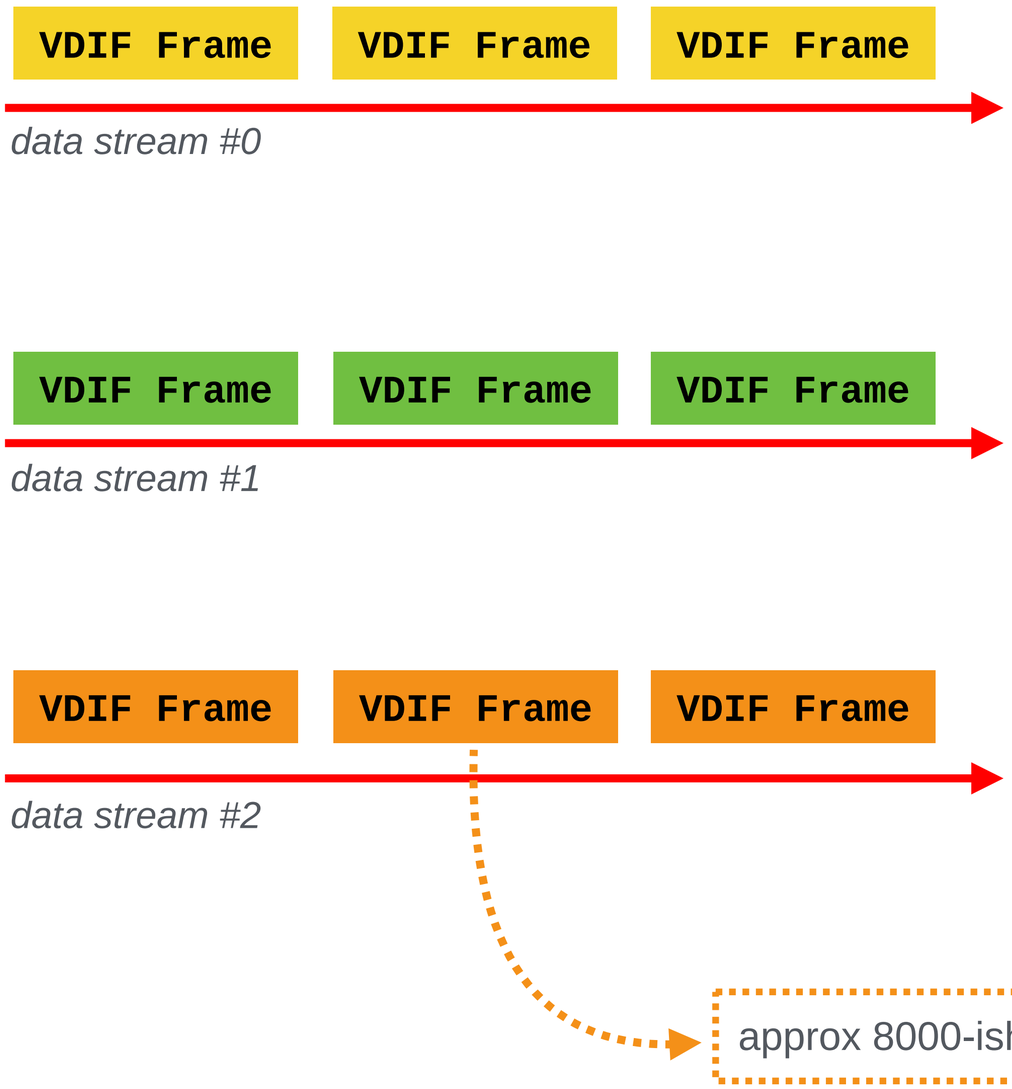
VLBI data recorder(s)

end(s)

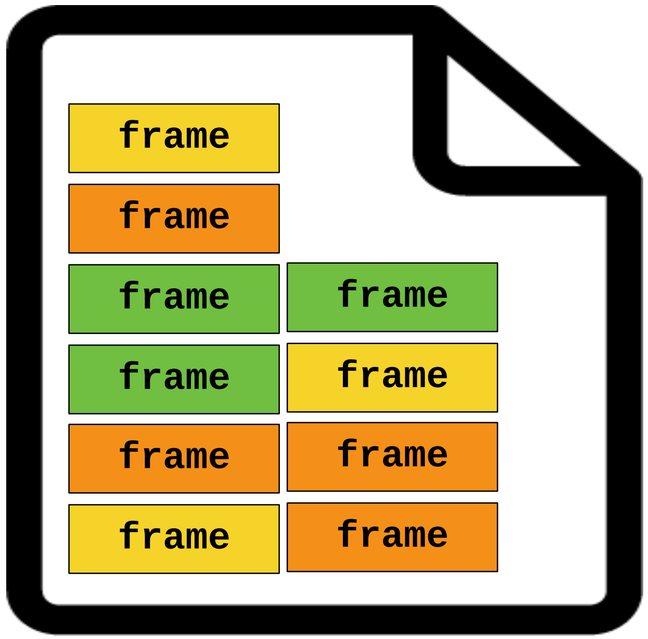


VLBI

end(s)


VDIF Frame
VDIF Frame
VDIF Frame
data stream #0
VDIF Frame
VDIF Frame
VDIF Frame
data stream #1
VDIF Frame
VDIF Frame
VDIF Frame
data stream #2
approx 8000-ish bytes / frame

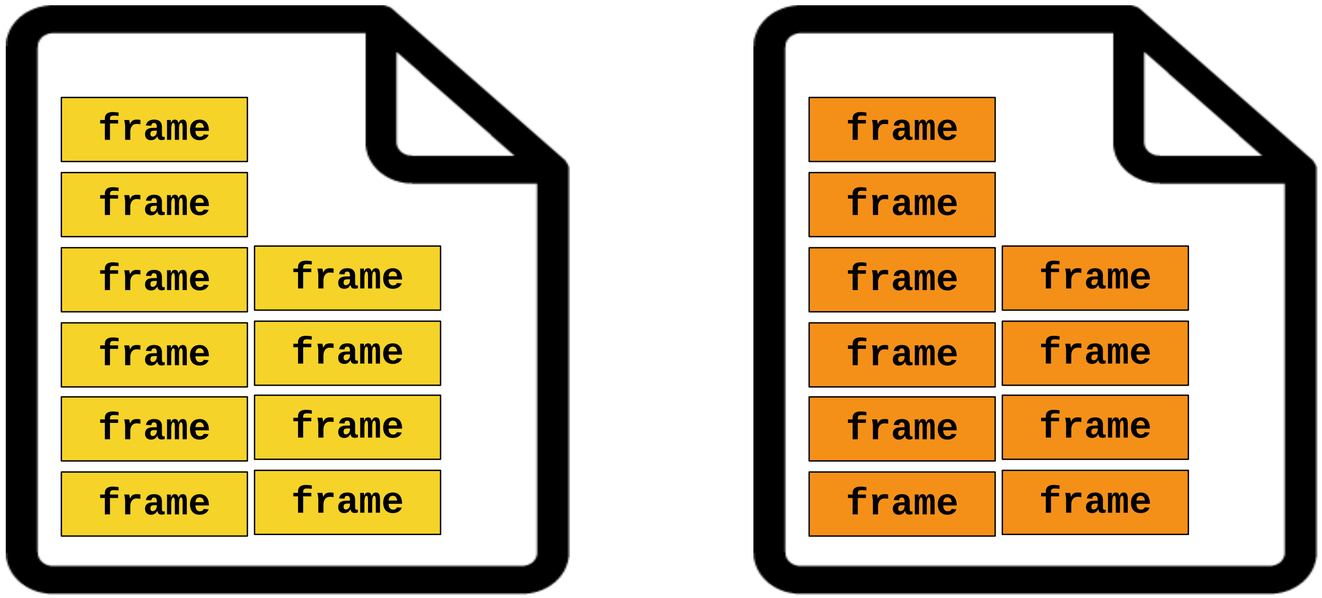
VLBI



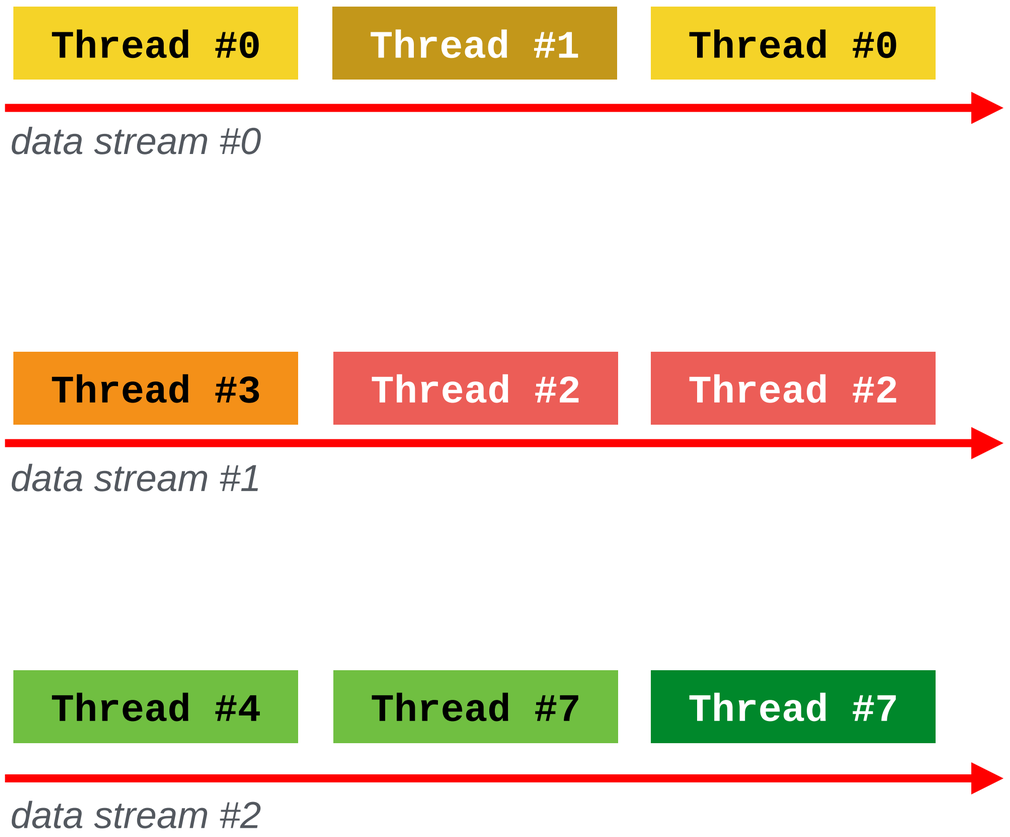
EXP_STA_SCAN(vbs/mk6)

```
net_port = [ip1@]port1[=suffix1]:
           [ip2@]port2[=suffix2];
```

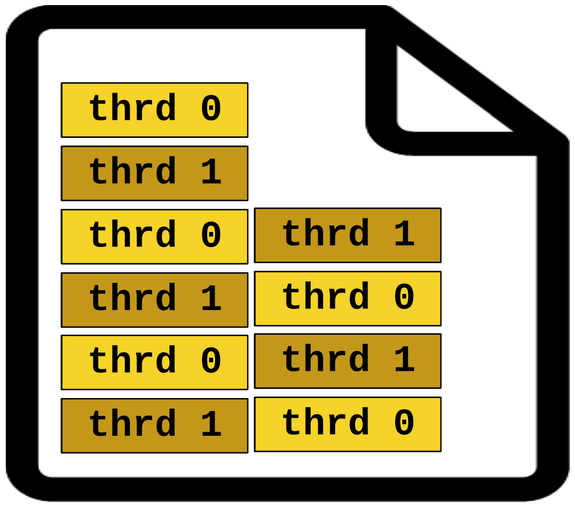
```
net_port = [ip1@]port1[=suffix1]:
           [ip2@]port2[=suffix2];
  record = on : <scan> ;
```


frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
EXP_STA_SCAN_ds<suffix1>
EXP_STA_SCAN_ds<suffix2>
. . .

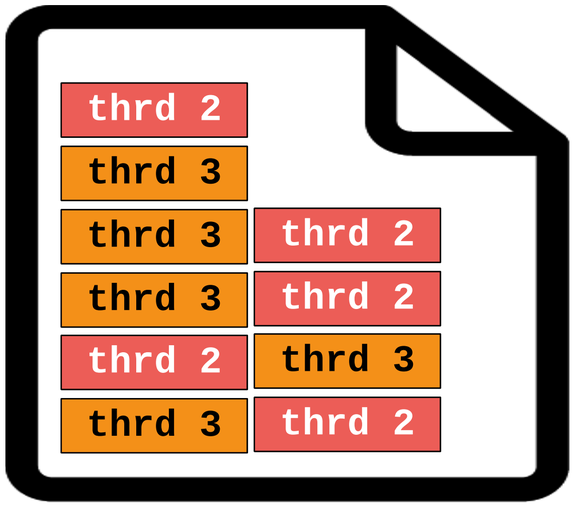
end(s)


Thread #0
Thread #1
Thread #0
data stream #0
Thread #3
Thread #2
Thread #2
data stream #1
Thread #4
Thread #7
Thread #7
data stream #2

VLBI



EXP_STA_SCAN_ds<suffix1>



EXP_STA_SCAN_ds<suffix2>

. . .

The **datastream=** command

See this commit log for brief usage/help/explanation:
https://github.com/jive-vlbi/jive5ab/commit/74c3f1b35bd258f344def8a50cae805381420549

The standard 32-byte VDIF Data Frame Header is shown in Figure 3.

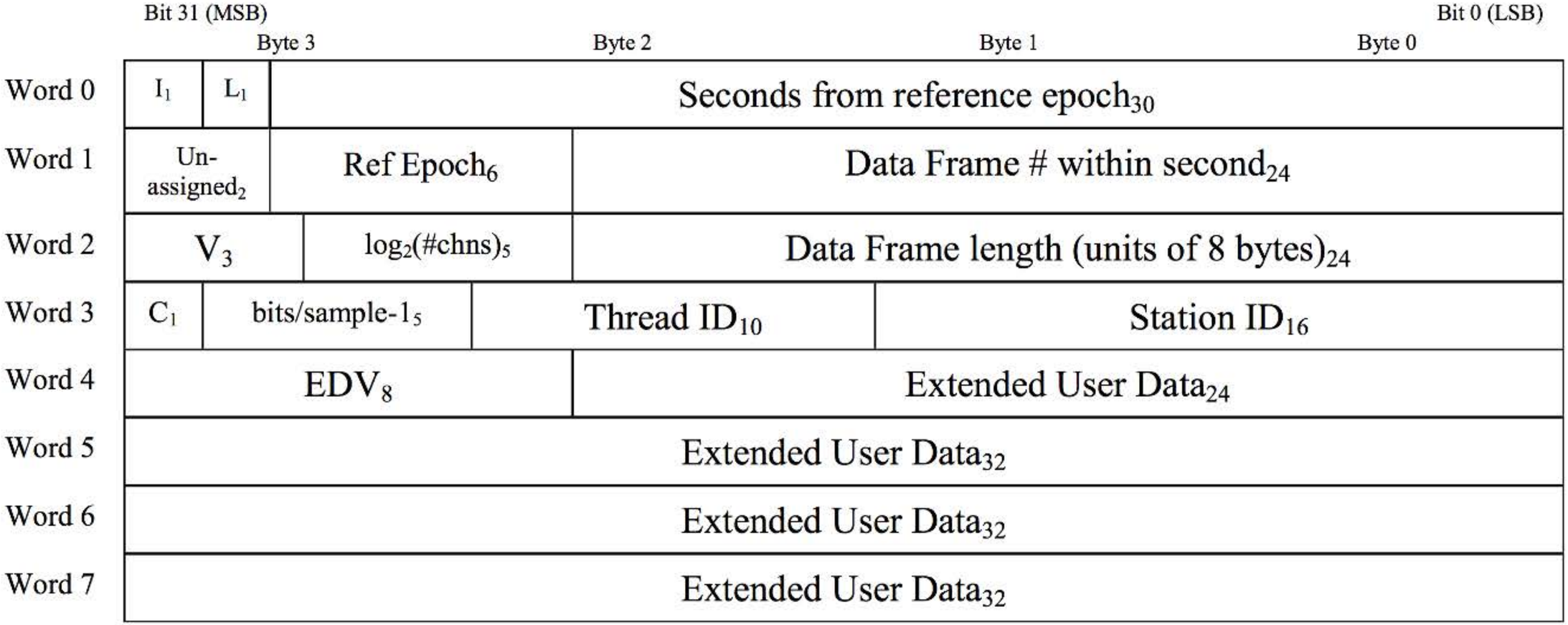


Figure 3: VDIF Data Frame Header format; subscripts are field lengths in bits; byte #s indicate relative byte address within 32-bit word (little endian format)

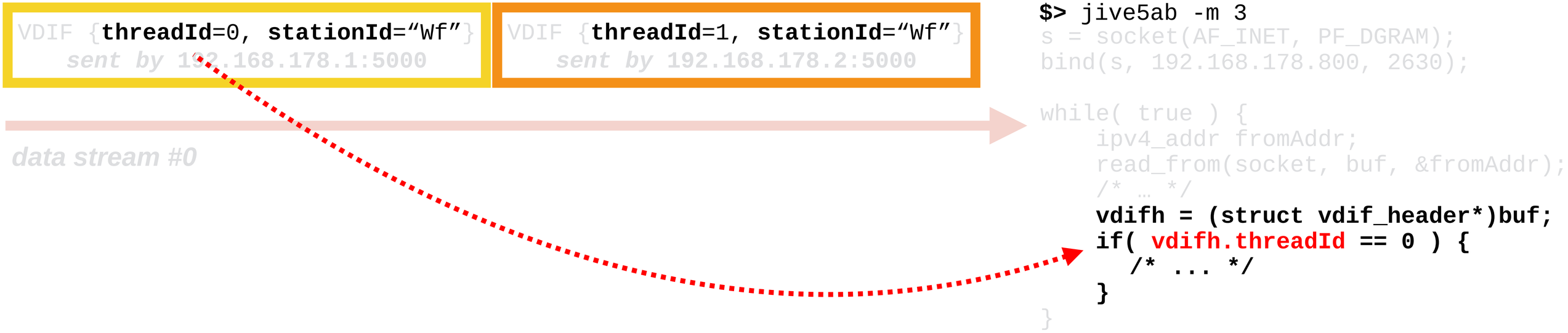
From VLBI Data Interchange Format (2009) - https://vlbi.org/vlbi-standards/vdif/

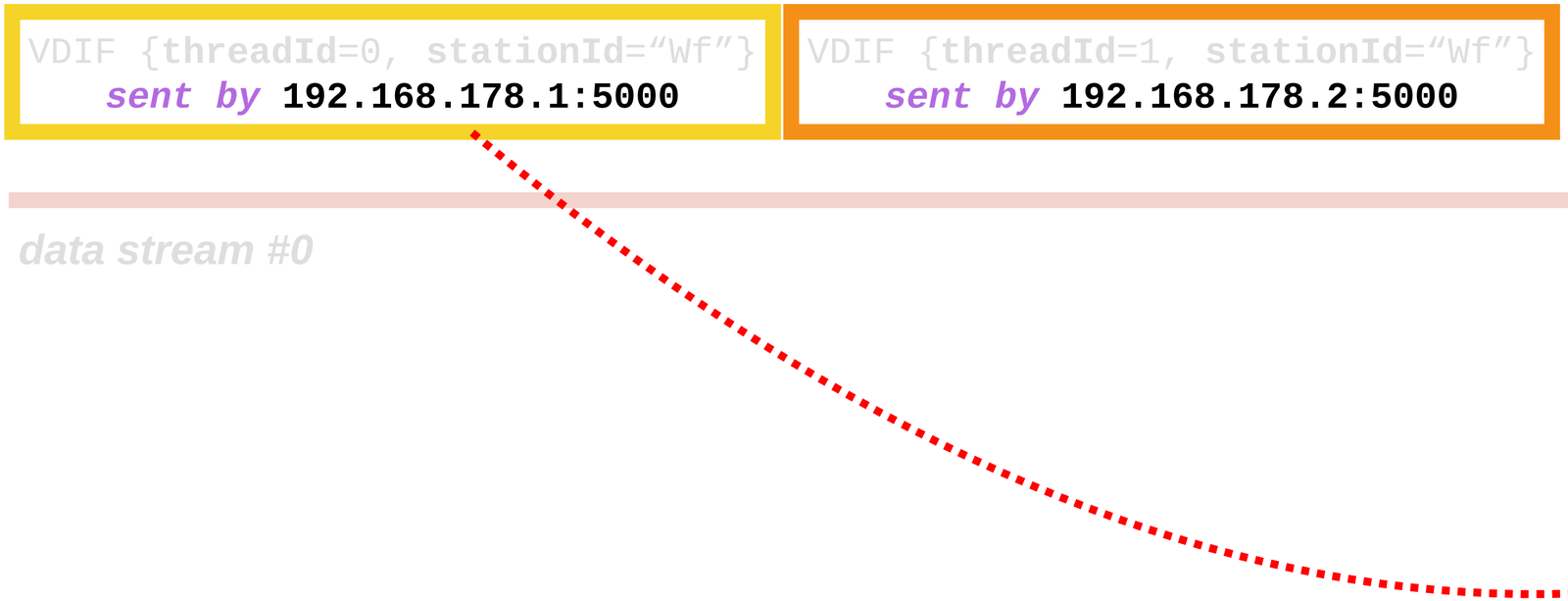
VDIF {**threadId**=0, **stationId**=“Wf”}
sent by **192.168.178.1:5000**

VDIF {**threadId**=1, **stationId**=“Wf”}
sent by **192.168.178.2:5000**

data stream #0

```
$> jive5ab -m 3
s = socket(AF_INET, PF_DGRAM);
bind(s, 192.168.178.800, 2630);

while( true ) {
    ipv4_addr fromAddr;
    read_from(socket, buf, &fromAddr);
    /* … */
    vdifh = (struct vdif_header*)buf;
    if( vdifh.threadId == 0 ) {
      /* ... */
    }
}
```

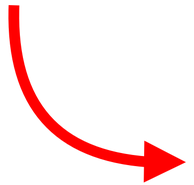

VDIF {threadId=0, stationId="Wf"}
sent by 192.168.178.1:5000
VDIF {threadId=1, stationId="Wf"}
sent by 192.168.178.2:5000
data stream #0
$> jive5ab -m 3
s = socket(AF_INET, PF_DGRAM);
bind(s, 192.168.178.800, 2630);
while(true) {
ipv4_addr fromAddr;
read_from(socket, buf, &fromAddr);
/* … */
vdifh = (struct vdif_header*)buf;
if(vdifh.threadId == 0) {
/* ... */
}
}




```
$> jive5ab -m 3
s = socket(AF_INET, PF_DGRAM);
bind(s, 192.168.178.800, 2630);

while( true ) {
    ipv4_addr fromAddr;
    read_from(socket, buf, &fromAddr);
    /* … */
    vdifh = (struct vdif_header*)buf;
    if( vdifh.threadId == 0 ) {
      /* ... */
    }
}
```

```
datastream = add : <suffix> : <match specification> [: <more matches> ];
```

```
datastream = add : <suffix> : <match specification> [: <more matches> ];
```

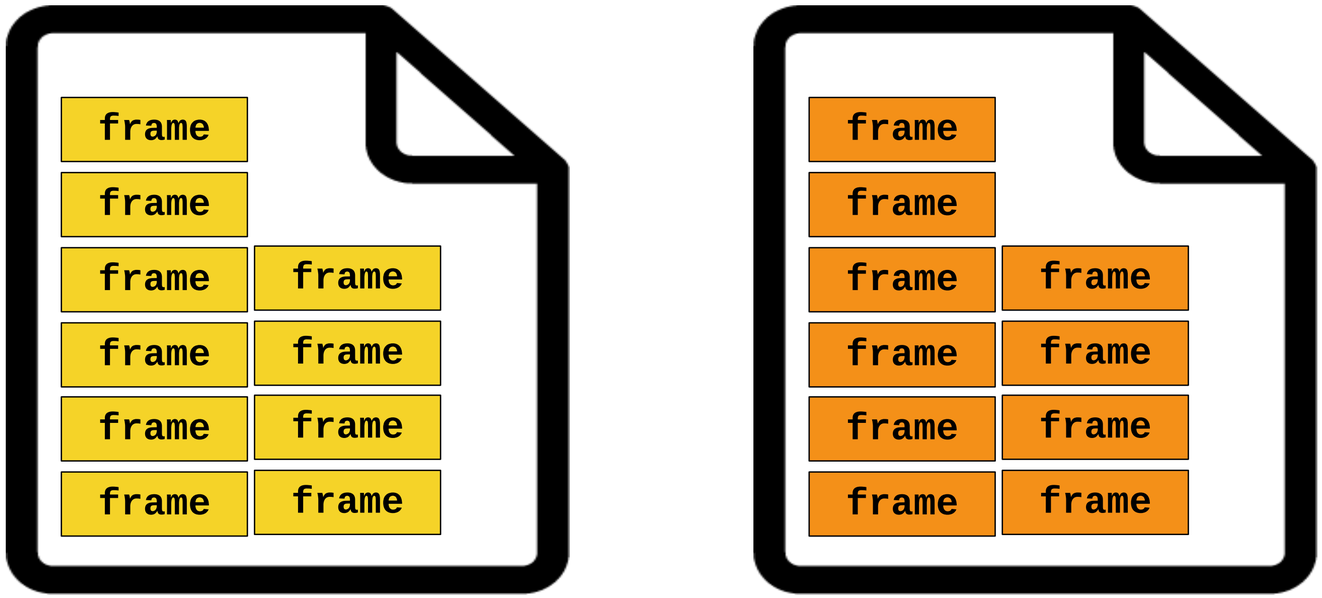


Match on any/all of:
- source IPv4
- source port
- VDIF threadId
- VDIF stationId

See this commit log for brief usage/help/explanation:
https://github.com/jive-vlbi/jive5ab/commit/74c3f1b35bd258f344def8a50cae805381420549

```
datastream = add : <suffix> : <match specification> [: <more matches> ];
```

See this commit log for brief usage/help/explanation:
https://github.com/jive-vlbi/jive5ab/commit/74c3f1b35bd258f344def8a50cae805381420549


frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
frame
EXP_STA_SCAN_ds<suffix1>
EXP_STA_SCAN_ds<suffix2>
. . .

Thanks for attention!

```
00:53:20;mk5=record? nthread
00:53:20/mk5/!record? : 4 : 32 ;

00:53:31; mk5=net_protocol?
00:53:31/mk5/net protocol? 0 : udpsnor: 768000000 : 128000000 : 16 ;

00:53:39; mk5=net_port?
00:53:39/mk5/!net_port? 0 : 2630 : 2631 : 2632 : 2634 ;
```

```
00:53:20;mk5=record? nthread
00:53:20/mk5/!record? : 4 : 32 ;

00:53:31; mk5=net_protocol?
00:53:31/mk5/net protocol? 0 : udpsnor: 768000000 : 128000000 : 16 ;

00:53:39; mk5=net_port?
00:53:39/mk5/!net_port? 0 : 2630 : 2631 : 2632 : 2634 ;
```

```
00:53:20;mk5=record? nthread
00:53:20/mk5/!record? : 4 : 32 ;

00:53:31; mk5=net_protocol?
00:53:31/mk5/net protocol? 0 : udpsnor: 768000000 : 128000000 : 16 ;

00:53:39; mk5=net_port?
00:53:39/mk5/!net_port? 0 : 2630 : 2631 : 2632 : 2634 ;
```

```
2024.009.15:46:35.17/mk5=version?2024.009.15:46:35.17/mk5/!version? 0 : jive5ab :
                        3.1.0-dev : 64bit : Debug :
                        koll : 08-Jan-2024 :
                        12h26m00s : nossapi :
                        /usr/local/src/etransfer ;

2024.009.15:48:37.54;mk5=scan_check?
2024.009.15:48:37.59/mk5/scan_check? 0 :?:
                        xxx_xx_no0001 : VDIF : 128 :
                        2024y009d15h43m13.3117s :
                        5.9812s : 256Mbps :
                        1060224 : 8032 ;

2024.009.15:48:42.56;mk5=evlbi?
2024.009.15:48:42.56/mk5/evlbi? 0 : total: 3062244:
                        loss : 0 (0.00%) :
                        out-of-order: 0 ( 0.00%) :
                        extent: 0 seqnr/pkt ;
```

```
2024.009.15:46:35.17/mk5=version?2024.009.15:46:35.17/mk5/!version? 0 : jive5ab :
                        3.1.0-dev : 64bit : Debug :
                        koll : 08-Jan-2024 :
                        12h26m00s : nossapi :
                        /usr/local/src/etransfer ;

2024.009.15:48:37.54;mk5=scan_check?
2024.009.15:48:37.59/mk5/scan_check? 0 :?:
                        xxx_xx_no0001 : VDIF : 128 :
                        2024y009d15h43m13.3117s :
                        5.9812s : 256Mbps :
                        1060224 : 8032 ;

2024.009.15:48:42.56;mk5=evlbi?
2024.009.15:48:42.56/mk5/evlbi? 0 : total: 3062244:
                        loss : 0 (0.00%) :
                        out-of-order: 0 ( 0.00%) :
                        extent: 0 seqnr/pkt ;
```

```
2024.009.15:46:35.17/mk5=version?2024.009.15:46:35.17/mk5/!version? 0 : jive5ab :
                        3.1.0-dev : 64bit : Debug :
                        koll : 08-Jan-2024 :
                        12h26m00s : nossapi :
                        /usr/local/src/etransfer ;

2024.009.15:48:37.54;mk5=scan_check?
2024.009.15:48:37.59/mk5/scan_check? 0 :?:
                        xxx_xx_no0001 : VDIF : 128 :
                        2024y009d15h43m13.3117s :
                        5.9812s : 256Mbps :
                        1060224 : 8032 ;

2024.009.15:48:42.56;mk5=evlbi?
2024.009.15:48:42.56/mk5/evlbi? 0 : total: 3062244:
                        loss : 0 (0.00%) :
                        out-of-order: 0 ( 0.00%) :
                        extent: 0 seqnr/pkt ;
```

128 x 256 Mbps = 32 Gbps
on a standard Mark6
with
32 x 10 TB HDD (or larger)