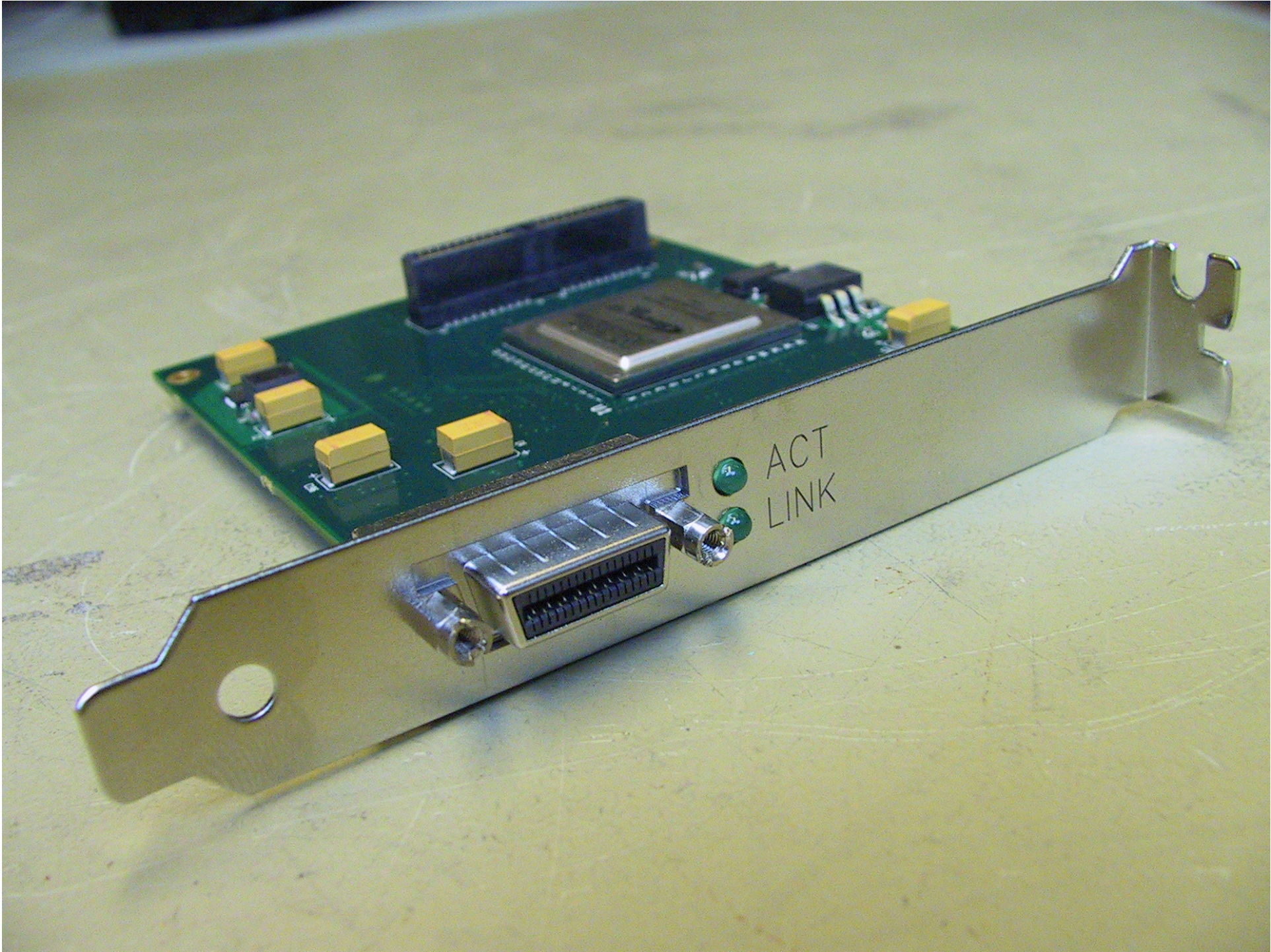


The Mark5C recorder

Harro Verkouter





The Mark5 family of recorders

	A	A+	B	B+	C
interface	RS422	RS422	VSI-H	VSI-H	10Gbps ethernet (CX4)
datasource	Mark4/VLBA formatter		DBBC R1002		DBBC+FiLa10G R1002+? RDBE
rec bandwidth	≤1Gbps	id.	≤1Gbps	≤2Gbps	≤4Gbps
e-VLBI bandwidth	≤1Gbps	≤1Gbps	≤1Gbps	≤1Gbps	(≤1Gbps)
why “+”		play 5B		≤2Gbps	
control s/w	Mark5A jive5ab		dimino jive5ab		drs (jive5ab)
timestamping	formatter		5B/+ itself		FiLa10G RDBE
1PPS needed?	no		yes		no

fully supported/verified
not/partially supported

Theory of operation

- It records (a part of) networkpackets
 - there is no intrinsic Mark5C format
- datasource *must* do (realtime) start/stop of dataflow
 - **the Mark5C can not stop a 4Gbps recording!**
- dual or single bank mode
 - 1 diskpack = single bank \leq 2Gbps
 - 2 diskpacks = dual bank \leq 4Gbps
 - two diskpacks become a single VSN
 - cannot share diskmodules between the two!
- support for
 - ethernet MAC address filtering
 - packet length filtering
 - optional reordering/fillpattern (payload contains PSN)

Mark5C 10Gbps IPv4 configuration

- none!
- no ethernet MAC address
- no IPv4 address
- no IPv4 protocol stack
 - does not respond to ARP requests
 - no ping
- not visible as eth<*n*> from Linux
 - no /sbin/ifconfig

Connecting the Mark5C

The zero intelligence req' d version:



RDBE

+



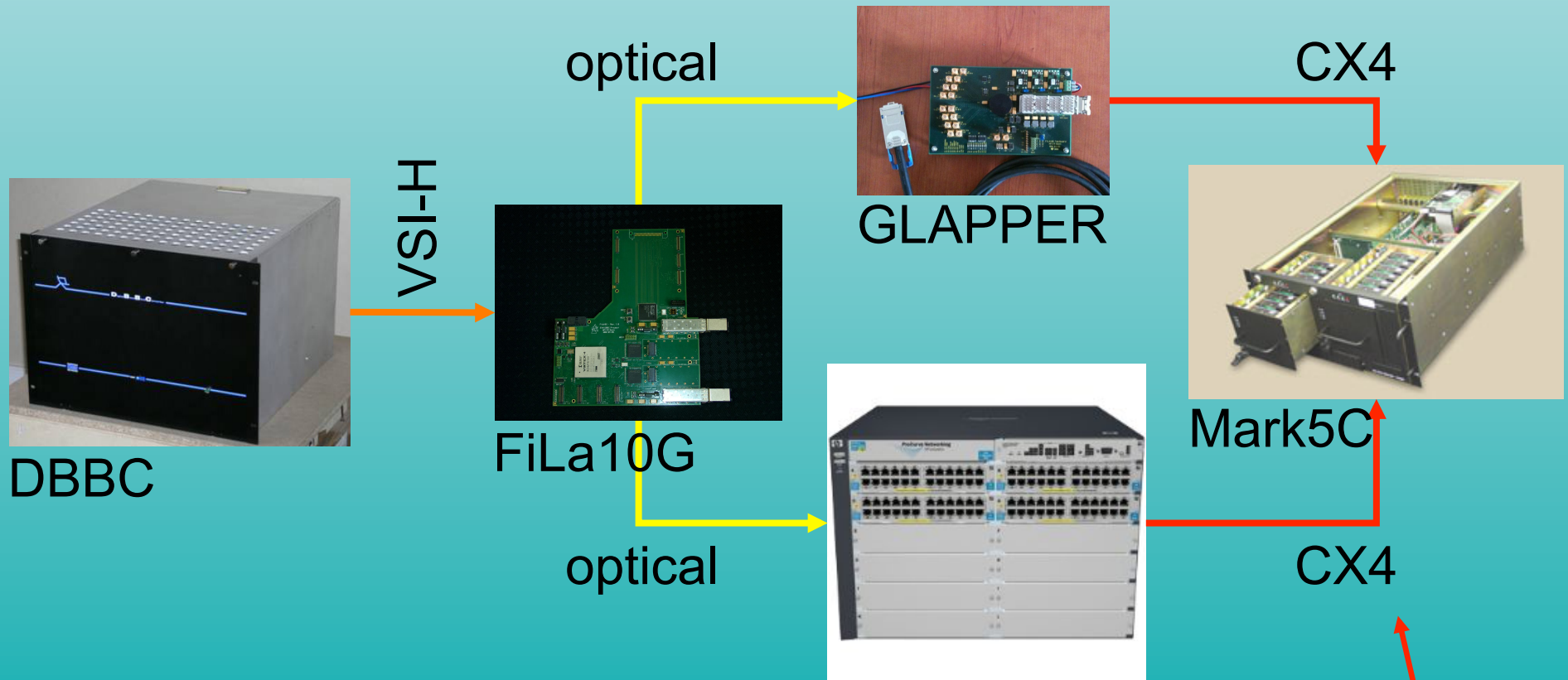
+



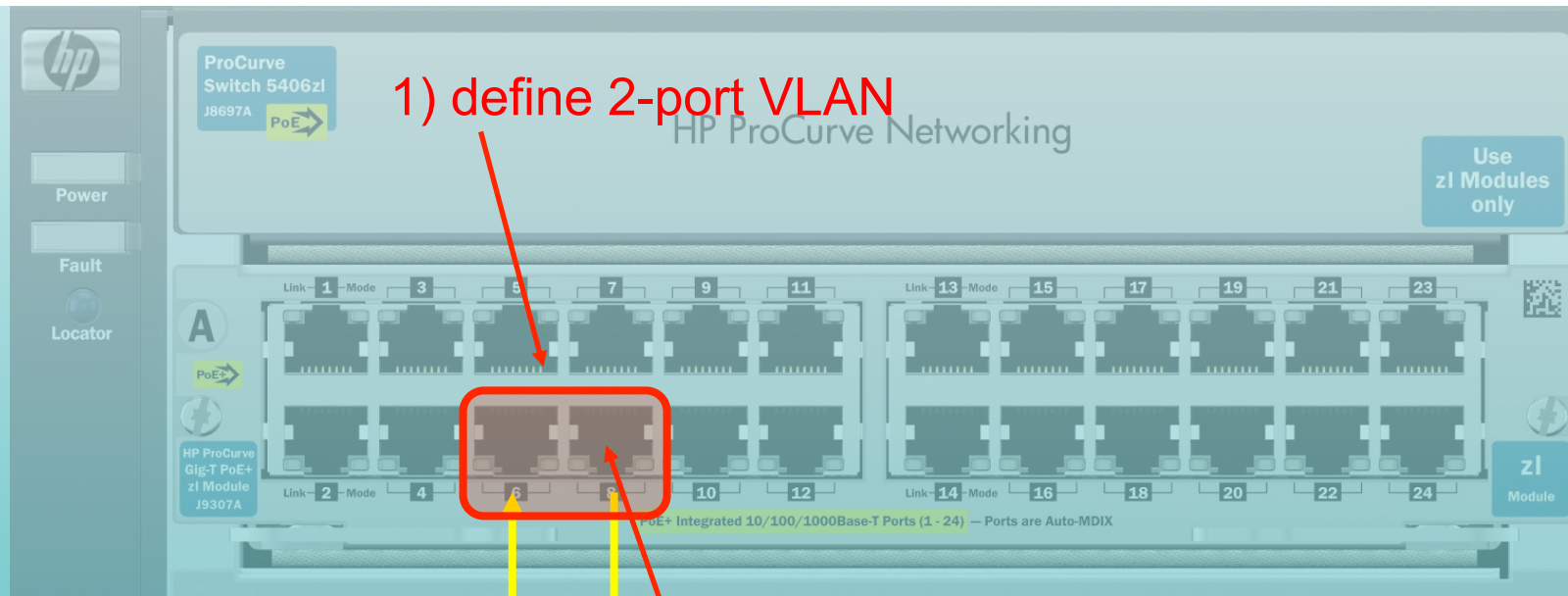
Mark5C

maximum of 4 meters CX4!

Connecting the DBBC to Mark5C



again: maximum of 4 meters CX4!



1) define 2-port VLAN

2) invent MAC address for 5C, program statically on this port

3) program arbitrary IP-address + invented MAC into FiLa10G arp table



FiLa10G



Mark5C

Control software

- /usr/bin/drs - server
 - “data recording service”
 - Python + C++ module to interface with StreamStor
- /usr/bin/drs_client
 - user commandline interface
 - or use the Field System
- /usr/bin/edrs
 - to end drs server
- StreamStor utilities (separate install)
 - SSErase, SSRreset

Control software - in practice

- current drs is case sensitive
 - “VSN?”, “VSN=...”, “MAC_list”, “SS_rev”
 - fixed in next release
- **needs** semicolon “;”
 - from FS:
 - “mk5=VSN?” will time-out
 - “mk5=VSN?;” will work
 - drs_client adds semicolon silently

Personality and mode

- “drs” supports multiple personalities
 - “file:<dataroot>”
 - “mark5c:[dual]bank”
 - *must* be set
 - no default when /usr/bin/drs starts
- mode affects recorded scan postprocessing
 - “mark5b:<bitstreammask>:<decimation>”
 - “vdif”
 - “unk” (nown)

Large parameterspace: not all combinations yield good results.
For software correlators recommend:

“personality=mark5c:[dual]bank” + “mode=unk”

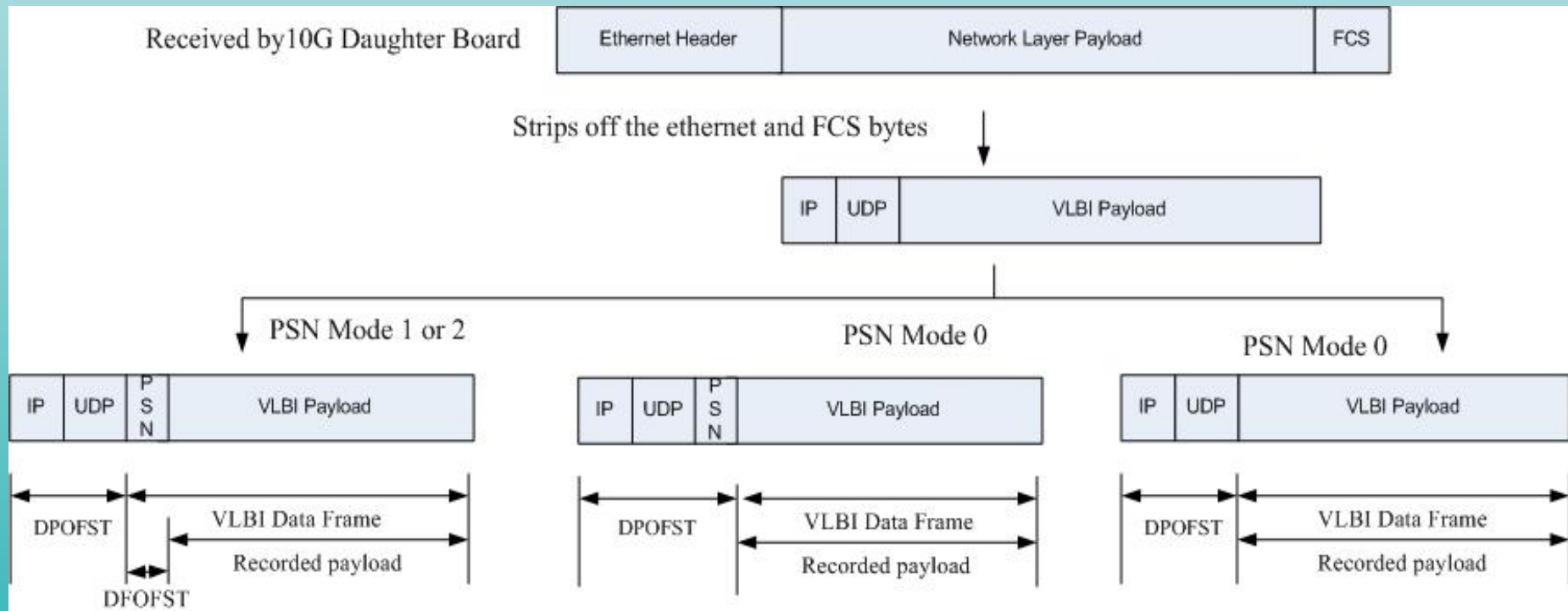
Setting up diskpacks (2Gbps)

- ensure no drs running (“/usr/bin/edrs”)
- insert diskpack in bank A and activate
 - important to use bank A!
 - make sure the diskpack *can* do 2Gbps
- /usr/bin/SSErase
- start /usr/bin/drs
- in another window start drs_client
 - “personality=mark5c:bank;”
 - “protect=off; VSN=<vsn>;”
 - “protect=off; reset=erase;”
 - execute twice (it’s one of those 5C things)

Setting up diskpacks (4Gbps)

- almost like 2Gbps ...
 - SSErase the two diskpacks individually *in bank A!* (with nothing in bank B)
 - then insert + activate both
- start /usr/bin/drs again
- in another window start drs_client
 - “personality = mark5c:dualbank;”
 - note the different personality
 - “protect=off; VSN=<vsn>;”
 - both diskpacks now become a single VSN
 - “protect=off; reset=erase;”
 - execute twice to force the userdirectory to be written

Packetrecording



20 8 8 1464 - 8964

packet = <DPOFST> : <DFOFST> : <length> : <PSN mode> : <PSNOFST> ;

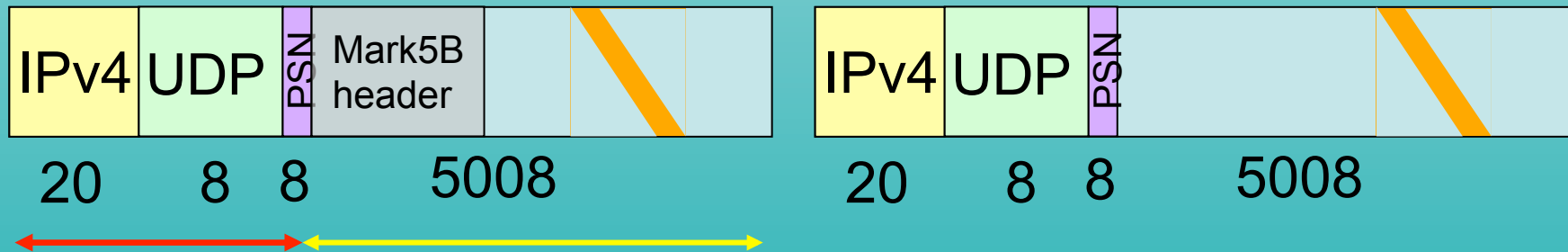
Image courtesy of C. Ruszczyk - "Mark5C User Manual"

PSN processing

- mode “0”
 - no PSN processing, recorded as on the network
- mode “1”
 - fix network reordering
 - insert fillpattern for missing packets
- mode “2”
 - like mode “1”, packet rejected if most significant bit is set (e.g. VDIF)

PSN modes “1” and “2” are not very well tested and may not work as advertised

Example: FiLa10G settings



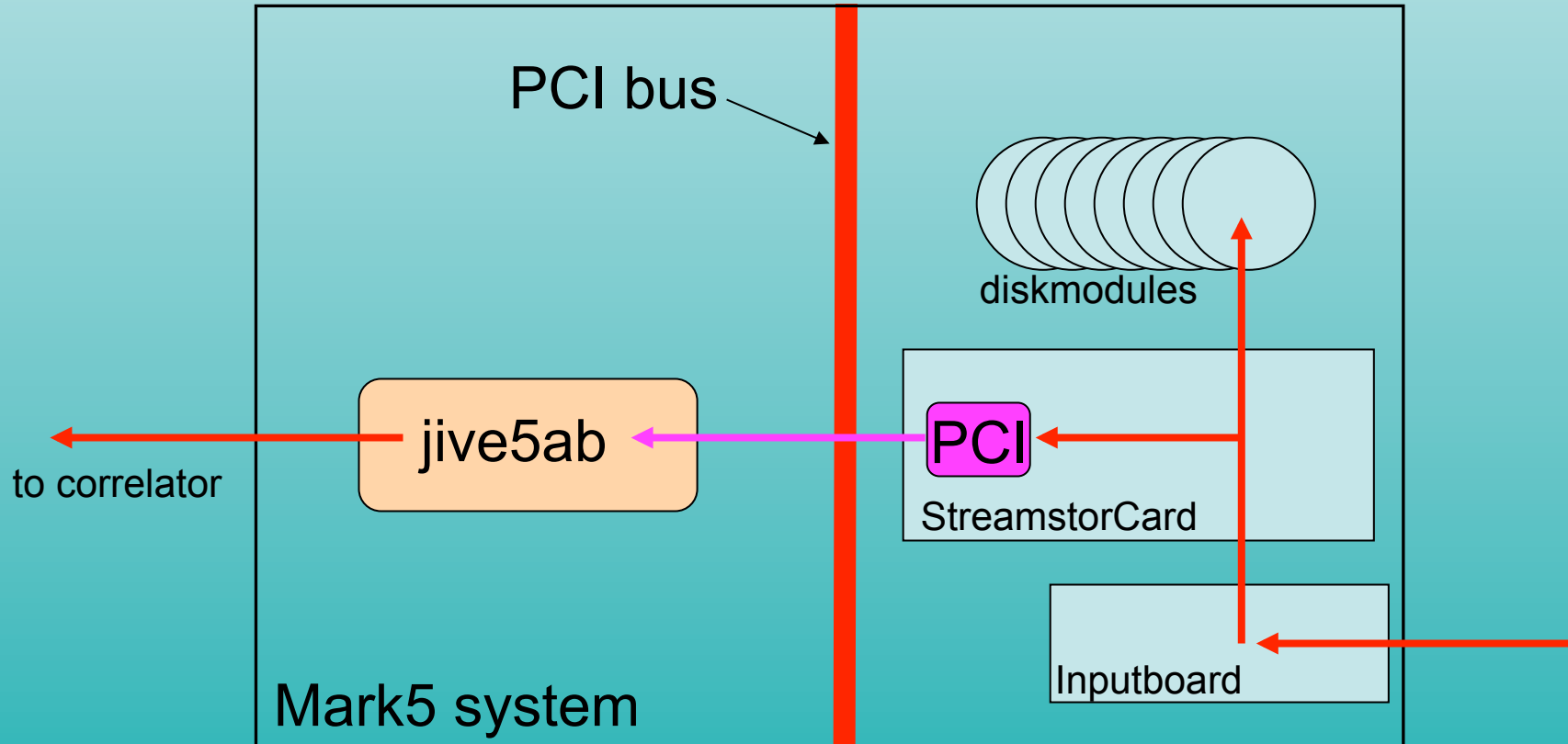
do not do PSN processing:
packet=36:0:5008:0:0

Summarizing: from 0 to 4Gbps in four commands

- > personality=mark5c:dualbank
prepare Mark5C recording mode
- > mode=unk
do not update metadata after end of recording
- > packet=36:0:5008:0:0
set appropriate packet characteristics
- > record=on:<scanname>
start the recording

Note: this assumes the diskpacks have been prepared as presented earlier

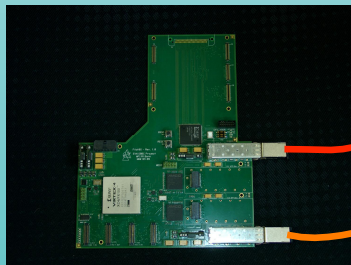
1Gbps e-VLBI limit



— high speed datapath ($\geq 4\text{Gbps}$)

— low speed component/datapath ($\sim 1.4\text{Gbps}$)

4Gbps e-VLBI with the harrobox



FiLa10G



Mark5C



Harrobox

- 8 cores @3.5Ghz
- 8GB RAM
- 1Gbps: RJ45
- 10Gbps: CX4, SFP+

- jive5ab
- cornerturning
- Mark5B to VDIF conversion
- distribution to >1 destination

to correlator

