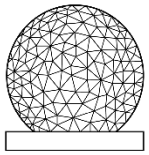
Mark6 Operations

10th IVS TOW Workshop

Chester “Chet” Ruszczyk

chester@mit.edu



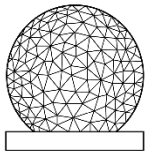
MIT
HAYSTACK
OBSERVATORY

Objective

- Mark6 General Information
- Mark6 Applications
- Disk Modules
- Recording
- Play Back
- Next Steps

Mark6 General Information

- Setup
  - Cabling for SAS controllers
  - Order is not critical but important
    - Why?
      - Individual disk information using the ***disk_info*** command is based upon certain order.
      - If a disk fails, poor performance there is not a one to one correspondence unless cabling is consistent.
      - You will have to determine it by probing additional disk_info states.
        - A disk detective



MIT
HAYSTACK
OBSERVATORY

Cabling for HBA Controller Cards

Version 2

Version 3

- Dependent on Version 2 vs. Version 3 HBA cards
  - Cable connectors are different
- Yellow / Red Dots to aid in connection cables
- We put stickers on the cables / disk modules
- If you do not use stickers there is a rule of thumb to follow
  - White label on cable is always on top
    - Represents the red dots

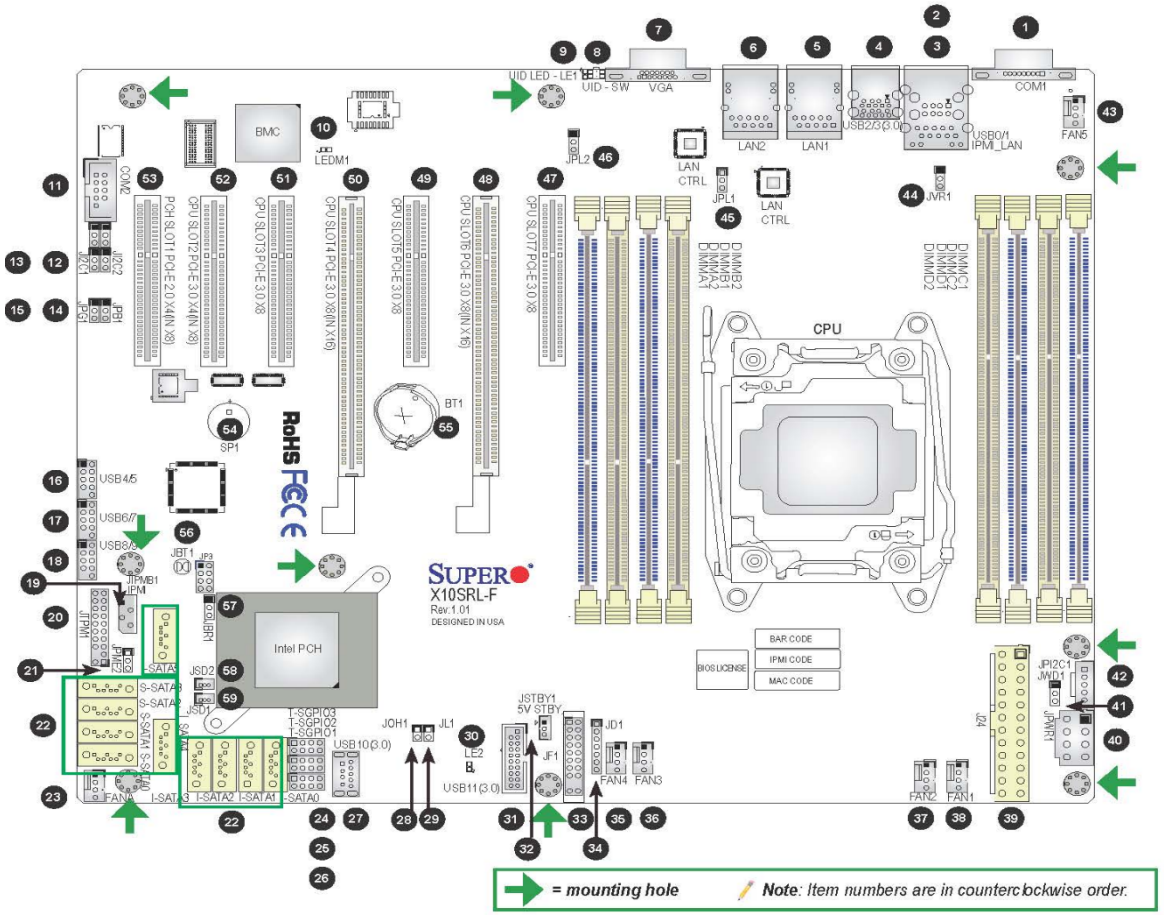
MIT HAYSTACK OBSERVATORY

Cable Connection

MIT
HAYSTACK
OBSERVATORY

MARK6 Motherboard Layout

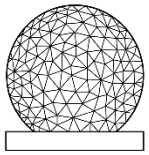
- Around 2016 MB update
  - Version 3 HBA cards
  - HBA – Slot 47, 48
  - 10G NICs – Slot 49,50
- Critical for 16Gbps recording



MIT HAYSTACK OBSERVATORY

On Boot Up

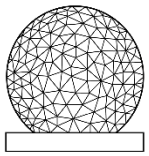
- SAS controller cards bios executes before motherboard bios
  - Enter and disable boot up from disks attached to Controllers.
    - Now if the system reboots with disk modules keyed on
    - It will not look for a master boot record on the disk modules
    - It will boot normally and not hang since no OS is found



MIT
HAYSTACK
OBSERVATORY

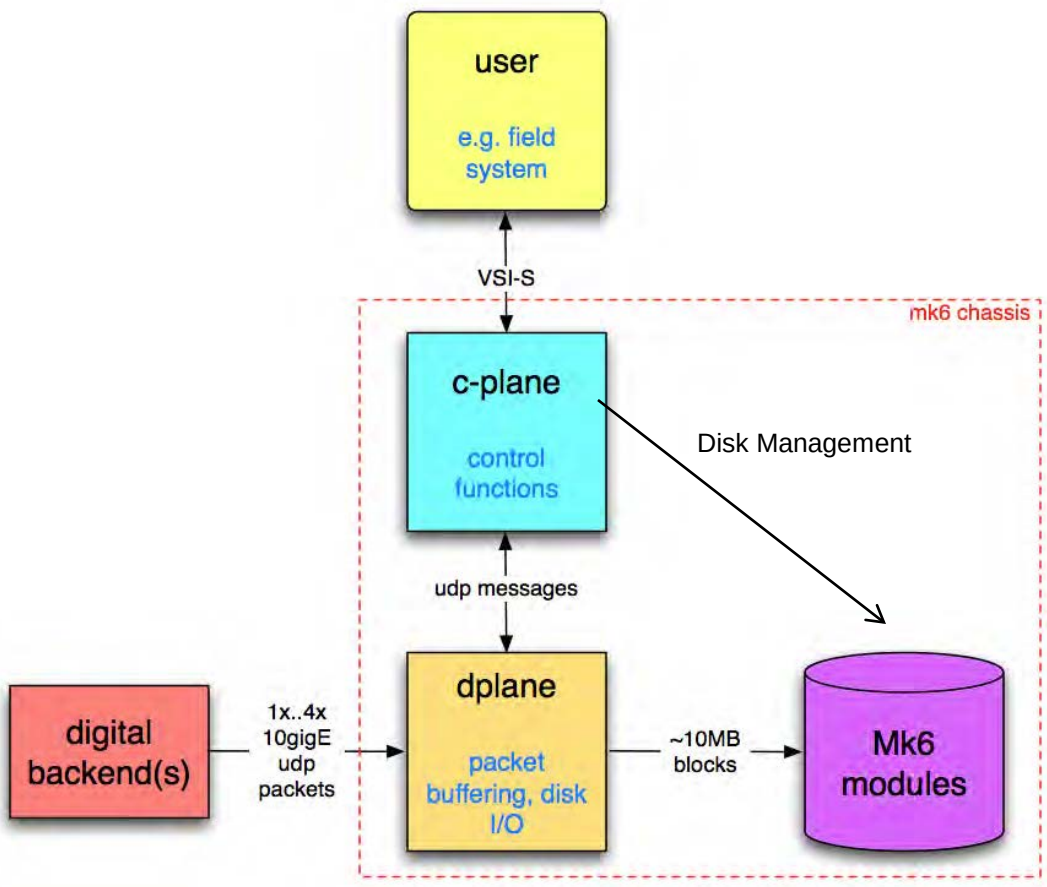
General

- Setup
  - Ethernet Interfaces
    - Eth0 - Eth5 do not exist? What is happening with my system?
    - OS disk was plugged in with different NIC cards
      - Linux assigned them eth0 - eth5
      - The new interfaces are eth6- eth11
  - How do I correct this?
    - As root remove the following file and reboot
      - rm /etc/udev/rules.d/70-persistent-net.rules
      - This file will be automatically regenerated on bootup



MIT
HAYSTACK
OBSERVATORY

Mark6 Software Architecture



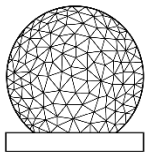
MIT HAYSTACK OBSERVATORY

Mark6 Applications

- cplane (control plane) application
  - 1.0.25 (geo) / 1.0.25 (astro)
- dplane (data plane) application
  - 1.22 (geo) / 1.22 (astro)
- End Stations
  - Need ***both*** applications / services to be running
- Correlators
  - Need only ***c-plane*** application / service is running

Mark6 Applications (cont)

- cplane / dplane can be started manually as before or now they can be started as services on bootup
- How to configure:
  - update-rc.d cplane default
  - update-rc.d dplane default
  - assumes latest versions of cplane / dplane files under /etc/init.d
- Configuration file
  - /etc/default/mark6 (Next slide)
    - Sets the Interrupts / smp affinity / CPU Cores
    - Critical for performance (recording)



MIT
HAYSTACK
OBSERVATORY

Configuration File

This file is sourced by /bin/sh from /etc/init.d/dplane

Defined in file /etc/default/mark6

Options to pass to mark6 which take effect with restart.

This specifies the ethernet ports to be used for incoming traffic.

(Up to 4 ports are supported; You **must** list only the ones actually to be used.)

MK6_OPTS=eth2:eth3:eth4:eth5

MK6_DRVR=myri10ge

Specifies the running directory--both planes log by default there.

MK6_RDIR=/var/log/mark6

dplane log level

MK6_DLOG=2

cdplane log level (Information, level 0 is debug)

MK6_CLOG=1

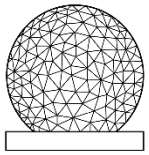
process umask

MK6_MASK=0002

MIT HAYSTACK OBSERVATORY

Mark6 Application (cont)

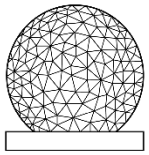
- Where are the log files?
  - /var/log/mark6
  - dplane-daemon log
  - cplane-daemon log
  - M6-2015-DOY-HH-MM-SS.log
- How to restart / stop / start when using services
  - sudo /etc/init.d/cplane {restart , stop, start }
  - sudo /etc/init.d/dplane {restart , stop, start }



MIT
HAYSTACK
OBSERVATORY

Disk Modules

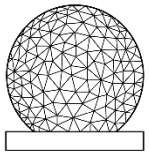
- Configured as RAID0 or scatter gather
  - RAID 5 coming for correlator operation
  - Recommend using scatter gather mode
- How to initialize a new module
  - mod_init = slot : number disks : MSN : sg : new
- How to remove a module from a group
  - mod_init = slot : number disks : MSN : sg : null
- How to erase
  - group = unprotect : slot
  - group = erase : slot



MIT
HAYSTACK
OBSERVATORY

Disk Modules (cont)

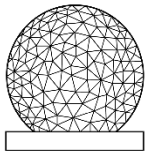
- Insert module in slot
- Connect cables
- Power -Turn key
  - Takes about 25 secs for module to be recognized by Linux kernel
    - Watch lights on module
  - Wait before querying on the module status
    - mstat ? all
    - mstat ? slot
- Requires 8 disks in module
  - cplane will not be happy with less
  - Note some say this is a bug, we say require good modules
    - Revisiting philosophy based on 2 years of operation



MIT
HAYSTACK
OBSERVATORY

Disk Modules (cont)

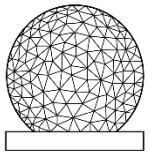
- Removing disks
  - group = close : slot
  - group = unmount : slot
    - Can verify using linux command df to see if modules are truly unmounted
  - turn key to remove power
  - query the module status
    - mstat ? all
    - mstat ? slot
  - Bug if you mstat? before turning off power
    - The meta data of disk 0 will be remounted



MIT
HAYSTACK
OBSERVATORY

Recording

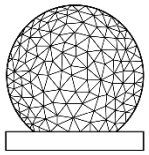
- Setup
  - input_stream command (next slide)
- Recording assumptions
  - Time is inspected in every header for all input streams defined
  - Only interfaces that are expecting data to be recorded should be defined
    - If a interface is defined and no data dplane will not close the files for it is expecting “ALL” streams specified to have valid data.
    - record=off must be issued to close files



MIT
HAYSTACK
OBSERVATORY

Recording

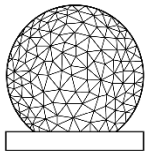
- Problems encountered
  - Data is not being recorded
    - input_streams declarations do not match data on wire
      - Use wireshark to capture a few packets and make sure
        - packet length and offsets are correct
    - vdif headers do not have proper time
      - dplane uses vdif time to determine how much data to record based on record command
    - vdif packets received have different reference epochs
      - dplane expects all streams to transmit the same reference epochs.



MIT
HAYSTACK
OBSERVATORY

Recording

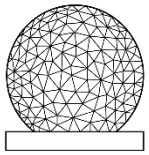
- Data is not being recorded (cont)
  - No data is being received on the interfaces
    - /sbin/ifconfig | grep -i “rx packets”
      - to see if the receive packet counters are incrementing
  - A group is not open for recording
- Why does cplane commands return two status fields?
  - The first is the vsi-s return code
  - The second is a cplane specific return code
    - Specified in command set
    - (see next slide)



MIT
HAYSTACK
OBSERVATORY

cplane return codes

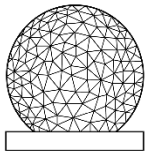
Mk6 return code	Command	Description
2		Specified group not open
10-19	delete	
20	execute	Invalid Action
21	execute	No filename provided
22	execute	Inconsistent filename used for append/finish process
23	execute	Duplicate filename
24	execute	Invalid upload sequence
25	execute	Attempted removal of non-existent xml file
30	group	Attempted open of multiple groups
31	group	Attempted open of incomplete group
32	group	‘unprotect’ not issued immediately before ‘erase’
33	group	‘auto’ option failed, only supports module types initialized as scatter / gather and not RAID
34	group	Attempted group open does not match subgroup defined in ‘input_stream’ configuration
40-49	gsm	
50-59	gsm_mask	
60	input_stream	Invalid subgroup declaration (group already open)
61	input_stream	Writing of subgroup meta data to disc failed
62	input_stream	Adding stream label failed, it already exists
63	input_stream	Specified stream label cannot be deleted it was not configured
64	input_stream	Committing configuration to dplane failed, not in an
65	input_stream	Commit failed, invalid sub-grouping compared to the open group_ref



MIT
HAYSTACK
OBSERVATORY

Recording (cont)

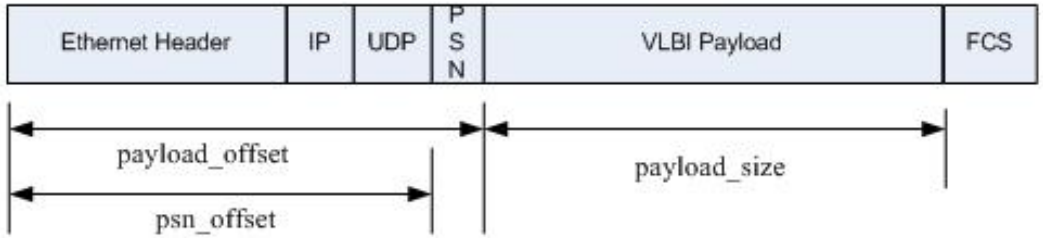
- Our data does not have PSN's how do I turn of checking?
  - set psn_offset to 0, this disables checking
- How can I check what vdif time is being received by dplane
  - use dpstat utility
  - turn on debug level logging on cplane and look at the log files
- Can you about a recording?
  - Yes, record=off
  - Will close any open files



MIT
HAYSTACK
OBSERVATORY

Mark6 Data Payload Definition and Parsing

Received by10G Ethernet NIC

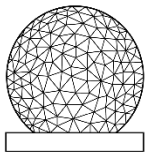


The "input_stream" command from the Mark6 command set specifies how to treat the incoming data on a specific Ethernet interface:

input_stream = <action> : <stream_label> : <data_format> : <payload_size> : <payload_offset> : <psn_offset> :
[<interface_ID>]: [<filter address>] :[<port>] : [<sub group ref>];

acton – {add, delete, commit}
delete with no stream label removes all labels defined
data_format – "m5b" for mark5B, and "vdif" for vdif VLBI payload format.
payload_size – VLBI Data Frame length in bytes, the length **must** be divisible by 8
payload_offset – number of bytes into the received packet to find the start of the VLBI Data Frame.
psn_offset – number of bytes into the received packet to find the start of the packet serial number
"0" represents no PSN in the incoming stream
"non-zero value" represents the location of the PSN in the incoming stream

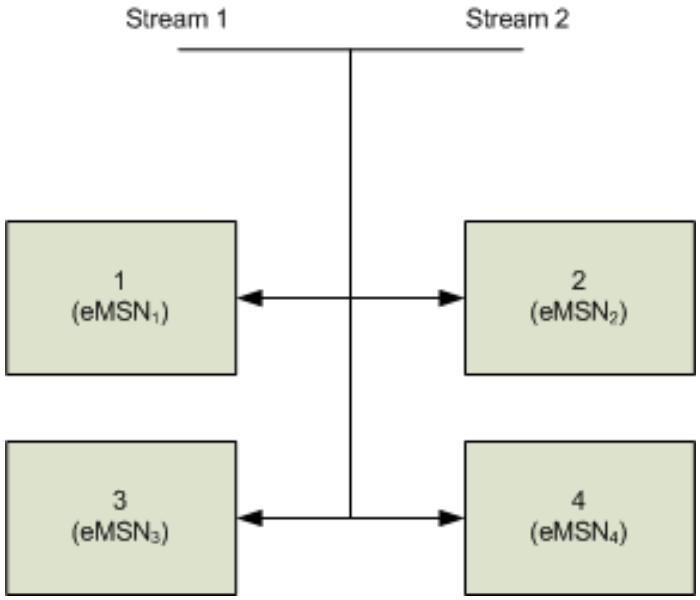
NOTE: Since the PSN can be the first word in the VLBI Data Frame or embedded in a VLBI header
(e.g. word 5 of the vdif header) specifies the number of bytes to locate the PSN.
Interface_id – {eth2, eth3, eth4, eth4, eth5}
Filter address and port not used
Sub group ref - Sub-group (of open group) to which this data stream "interface_ID" should be written to.



MIT
HAYSTACK
OBSERVATORY

Subgroup Feature

- Mark6 normal recording mode
  - group_ref = 1234
    - 4 disk modules open for recording
  - 2 input streams defined for receiving data
    - e.g. eth2, eth4
  - 8Gbps / input stream
  - 16 Gbps is written to all disk modules in group_ref



Subgroup Feature (cont)

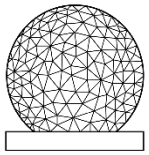
- Imagine if each Ethernet port receives a different polarization
  - eth2 <= RCP, eth4 <=LCP

- For existing Mark6 software if correlating a specific polarization, e.g. RCP
  - Requires all 4 disk modules to be inserted at correlator for processing.

Subgroup Feature (cont)

- If one disk module is lost in shipment both RCP and LCP are lost (25% of data lost).

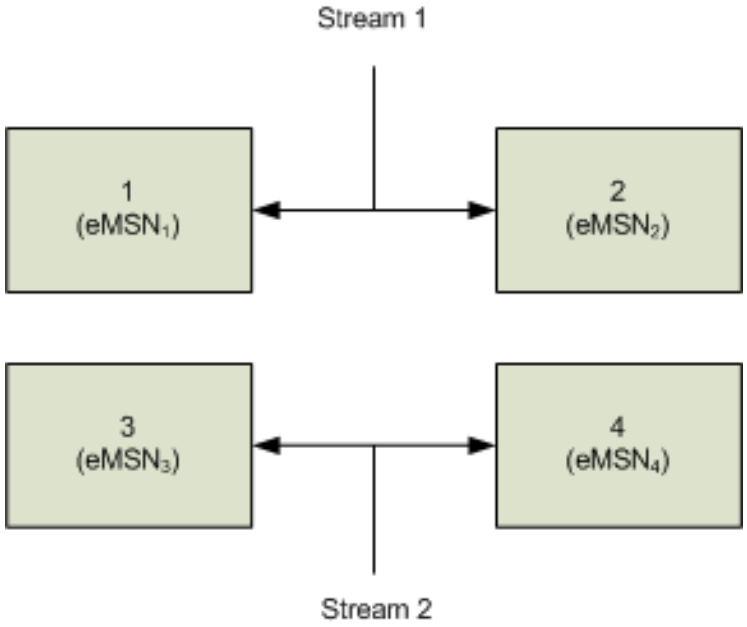
- The subgroup feature allows you to specify A specific input stream to be written to a “subgroup” of disk modules within the group_ref
  - granularity of 8 disks



MIT
HAYSTACK
OBSERVATORY

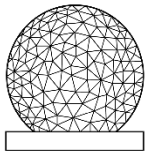
Subgroup Example

- group_ref = 1234
  - 4 disk modules open for recording
- input “Stream 1”
  - 8Gbps (RCP)
  - written to disk modules in slot 1 & 2
- input “Stream 2”
  - 8Gbps (LCP)
  - written to disk modules in slot 3 & 4



Subgroup Example (cont)

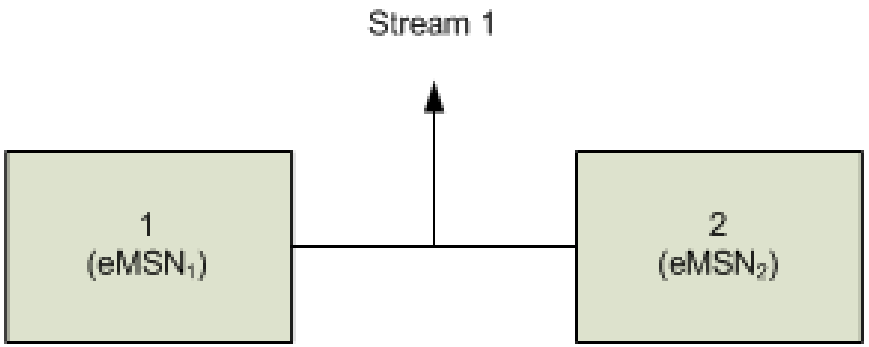
- When modules are at the correlator awaiting processing
  - RCP is scheduled for the participating antennas to be processed
    - Previously required all 4 disk modules
  - With subgrouping requires only disk modules that were written in Slot 1 & 2 be inserted at the correlator in a Mark6 correlator system
    - $eMSN_1$, $eMSN_2$
    - Do they have to be inserted into slots 1 & 2, **No**



MIT
HAYSTACK
OBSERVATORY

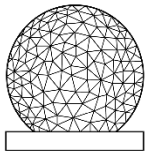
Subgroup Correlation

- RCP can now be processed.
  - Does not require all of group_ref
    - Only $eMSN_1$ and $eMSN_2$



Subgroup Restrictions

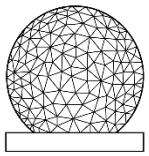
- Software c-plane restrictions
  - Once subgroups are defined, they must be kept for the group_ref when recording
    - No switching of subgroup's for the group, e.g.
      - input_stream 1 => 1,2
      - input_stream 2 => 3,4
      - record "n" scans
      - remove subgrouping as in "normal operations"
      - record "m" additional scans
      - **ILLEGAL**



MIT
HAYSTACK
OBSERVATORY

Subgroup Restrictions

- Software c-plane restrictions (cont)
  - Subgroup assignment must use all disk modules of open group
    - Example of illegal case:
      - group = open : 1234
      - input_stream 1 => 12
      - input_stream 2 => 3
      - disk in slot 4 not assigned : **ILLEGAL**
  - Complete subgroup modules are required for processing
    - input_stream 1 => 1,2 ($eMSN_1$, $eMSN_2$)
    - At correlator requires both $eMSN_1$, $eMSN_2$ inserted in same Mark6



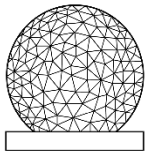
MIT
HAYSTACK
OBSERVATORY

Play Back

- Mount the disks
- group_members? slot
  - Number of disks in the group_ref
  - The associated disks eMSN in the group_ref
- When mounting, does order have to be preserved?
  - No you can place them in any slot of the Mark6's
- What about sub-grouped Modules?
  - Would need only complete subgroup to be mounted for data to be removed

Play Back (cont)

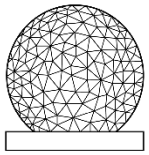
- Vdifuse (Geoff Crew)
  - Scatter / Gather Fuse interface for VDIF
    - Alma Phasing Project - verified
    - General purpose geodesy does not work
    - Mount Mark6 Modules with vdifuse
    - process the data directly from the disk modules to DiFX
- gator
  - Wrapper around gather and gather464 files
  - Gathers the scatter / gather files and creates a single file
  - Data must be ***dqa*** into separate threads for DiFX processing



MIT
HAYSTACK
OBSERVATORY

Play Back (cont)

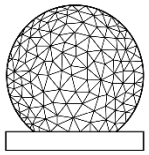
- gather
  - Gathers the scatter / gather files and creates a single file
  - VDIF payload characteristics
    - All threads are left as is (N threads)
    - If threadIDs are not unique they will be combined and cause errors
- gather464
  - Gathers the scatter / gather files and creates a single file
  - VDIF payload characteristics
    - All threads are merged into a single thread
    - Geodetic observations
      - 64 channels
  - Reduces correlation time by 4



MIT
HAYSTACK
OBSERVATORY

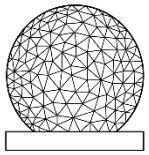
Mark6 Next Steps

- OS Upgrade path
  - NASA is requiring us to move to a RedHat based OS for 2020
  - Option of supporting both Debian and RH distributions but it is too difficult
  - June 2019 a CentOS 7 distribution of Mark6 will be made available to IVS community
    - Support thru 2024
- 10 G NIC un-obtainium
  - Myricom was purchased by private VC in 2018 and NIC cards can no longer be purchased
  - Spare evaluation has lead to Intel based devices
    - 82599ES based device
  - Evaluation of HotLava TAMbora 40G2S NIC is under way.



MIT
HAYSTACK
OBSERVATORY

Questions ?



MIT
HAYSTACK
OBSERVATORY