Correlation preparation

Aard Keimpema
keimpema@jive.eu

Configuration

- **V**LBI **Ex**periment (VEX) files
  - Describes how an observation was observed and recorded
  - Created by scheduling software (e.g. sched)
  - Is used by both correlator and observatories
  - However, VEX files created by sched don’t contain all information needed by the correlator
  - VEX 2.0 required for multiple data streams
  - Definition: https://vlbi.org/vlbi-standards/vex/

Tour of the VEX file: $MODE

```
$MODE;
def sess125.L1024;
   ref $IF = LO@1247MHzDPolNoTone:Wb;
   ref $IF = LO@1510MHzDPolNoTone:Ef;
   ref $IF = LO@1510MHzDPolNoTone#02:Hh;
   ref $IF = LO@186MHzDPolNoTone:Ur;
   ref $IF = LO@2272MHzDPolNoTone#02:Cm:Da:De:Kn:Pi;
   ref $IF = LO@730MHzDPolNoTone:O8;
   ref $BBC = 2BBCs:Cm:Da:De:Kn:Pi;
   ref $BBC = 8BBCs:Ef;
   ref $BBC = 8BBCs#02:Jb:Tr:Wb;
   ref $BBC = 8BBCs#03:Hh:O8;
   ref $BBC = 8BBCs#06:Ur;
   ref $BITSTREAMS = UrBitstreams:Ur;
   ref $THREADS = CmThreads:Cm:Da:De:Kn:Pi;
   ref $THREADS = EfThreads:Ef:O8:Wb;
   ref $FREQ = 1626.49MHz2x64MHz:Cm:Da:De:Kn:Pi;
   ref $FREQ = 1626.49MHz8x32MHz:Ef:O8:Ur:Wb;
   ref $PHASE_CAL_DETECT = NoDetect:Ef:O8:Ur:Wb;
   ref $PHASE_CAL_DETECT = NoDetect#02:Cm:Da:De:Kn:Pi;
   ref $PROCEDURES = Mode_01;
enddef;
```

- The mode block contains the basic setup of an observation.
- Not included in VEX files from (py)sched:
  - $BITSTREAMS (Mark5B)
  - $THREADS (VDIF)
- $THREADS is JIVE specific (VEX 1.5 only)
- In VEX 2.0 $DATASTREAMS is used for VDIF

Tour of the VEX file: $FREQ

```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Tour of the VEX file: $FREQ

```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Channel names, used internally by SFXC

Tour of the VEX file: $FREQ

```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Sky frequency of band edge

Frequencies referenced in SFXC output as
frequency number 0 (1626.49 MHz), and 1 (1690.49 MHz)

Tour of the VEX file: $FREQ

```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Bandwidth, notice they are not the same

Tour of the VEX file: $FREQ

```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Upper or lower sideband

Band from 1690.49 – 1658.49 MHz

Tour of the VEX file: $FREQ

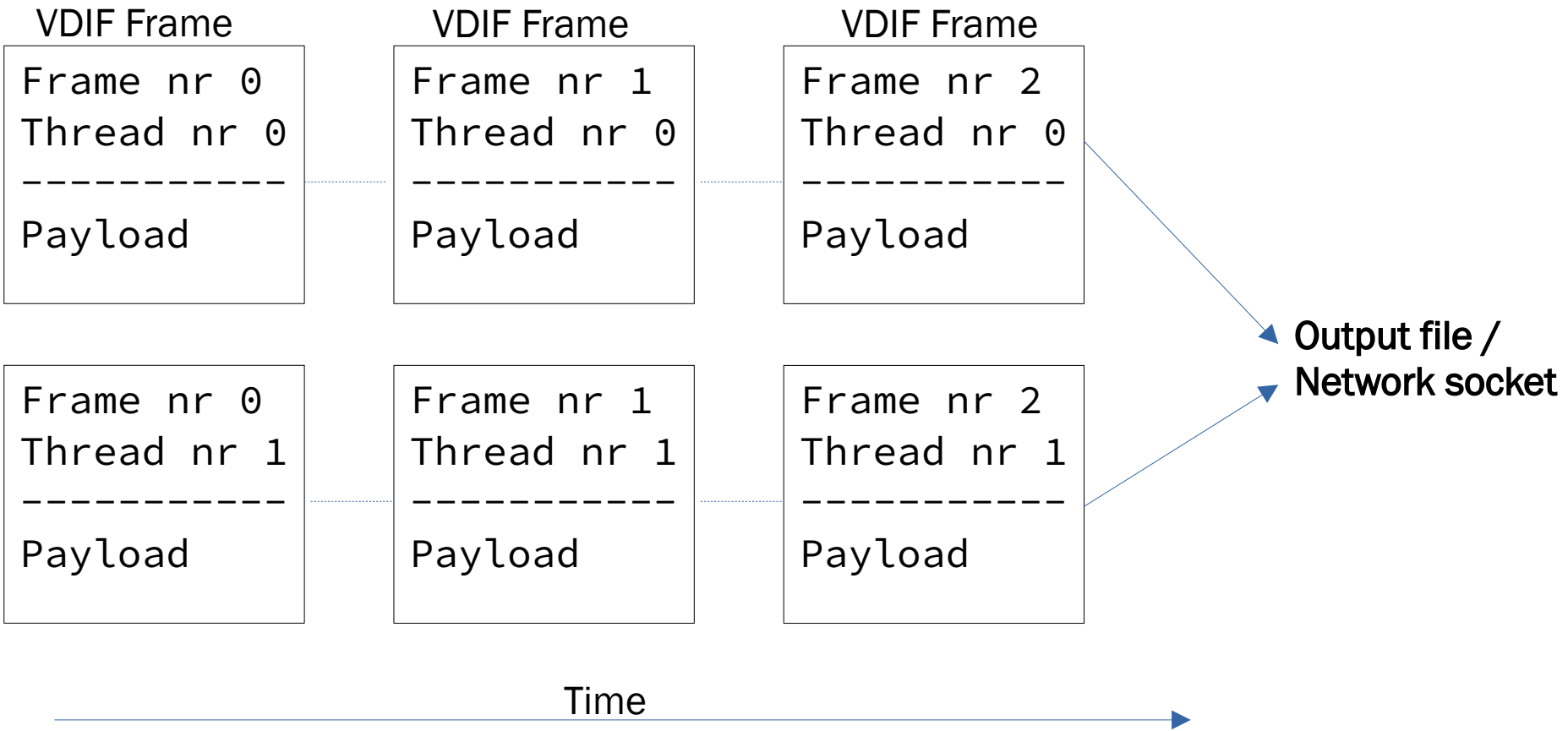
```
$FREQ;
def 1626.49MHz2x64MHz;
   sample_rate = 128.00000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 64.00000000 MHz : &CH02 : &BBC09 : &NoCal;
enddef;

def 1626.49MHz8x32MHz;
   sample_rate = 64.000000 Ms/sec;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH01 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : L : 32.00000000 MHz : &CH02 : &BBC09 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH03 : &BBC01 : &NoCal;
   chan_def =   : 1626.49000000 MHz : U : 32.00000000 MHz : &CH04 : &BBC09 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH05 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : L : 32.00000000 MHz : &CH06 : &BBC10 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH07 : &BBC02 : &NoCal;
   chan_def =   : 1690.49000000 MHz : U : 32.00000000 MHz : &CH08 : &BBC10 : &NoCal;
enddef;
```

Right Circular Polarisation (RCP)
Left Circular Polarisation (LCP)

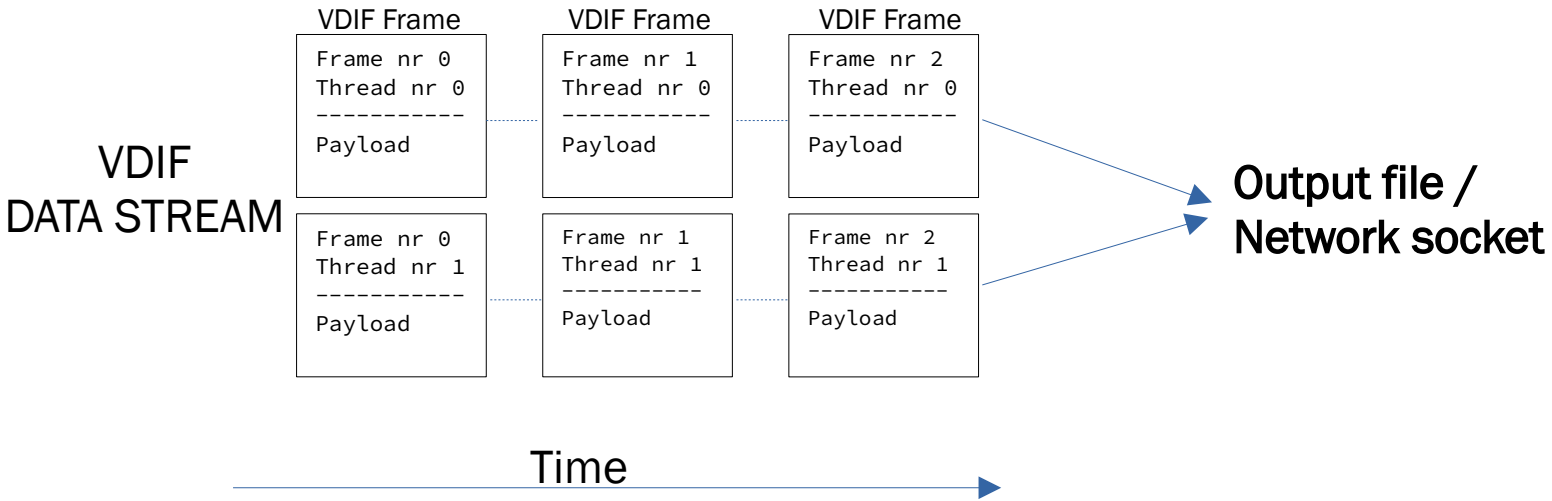
Data formats

Data format	VEX section	Description
Mark5a (Mark4 / VLBA)	$TRACKS	Obsolete, format originally meant for magnetic tape
Mark5B	$BITSTREAMS	Disk based format, mostly replaced by VDIF. Can do Max. 2 Gbps datarate, max. 2 bit quantisation, and only real valued data (no complex).
VDIF	$THREADS (VEX 1.5) $DATASTREAMS (VEX 2.0+)	De facto standard VLBI data format. - Up to 32 bit per sample - Complex sampling support - Can be split into multiple independent data streams (requires VEX 2) https://vlbi.org/vlbi-standards/vdif/

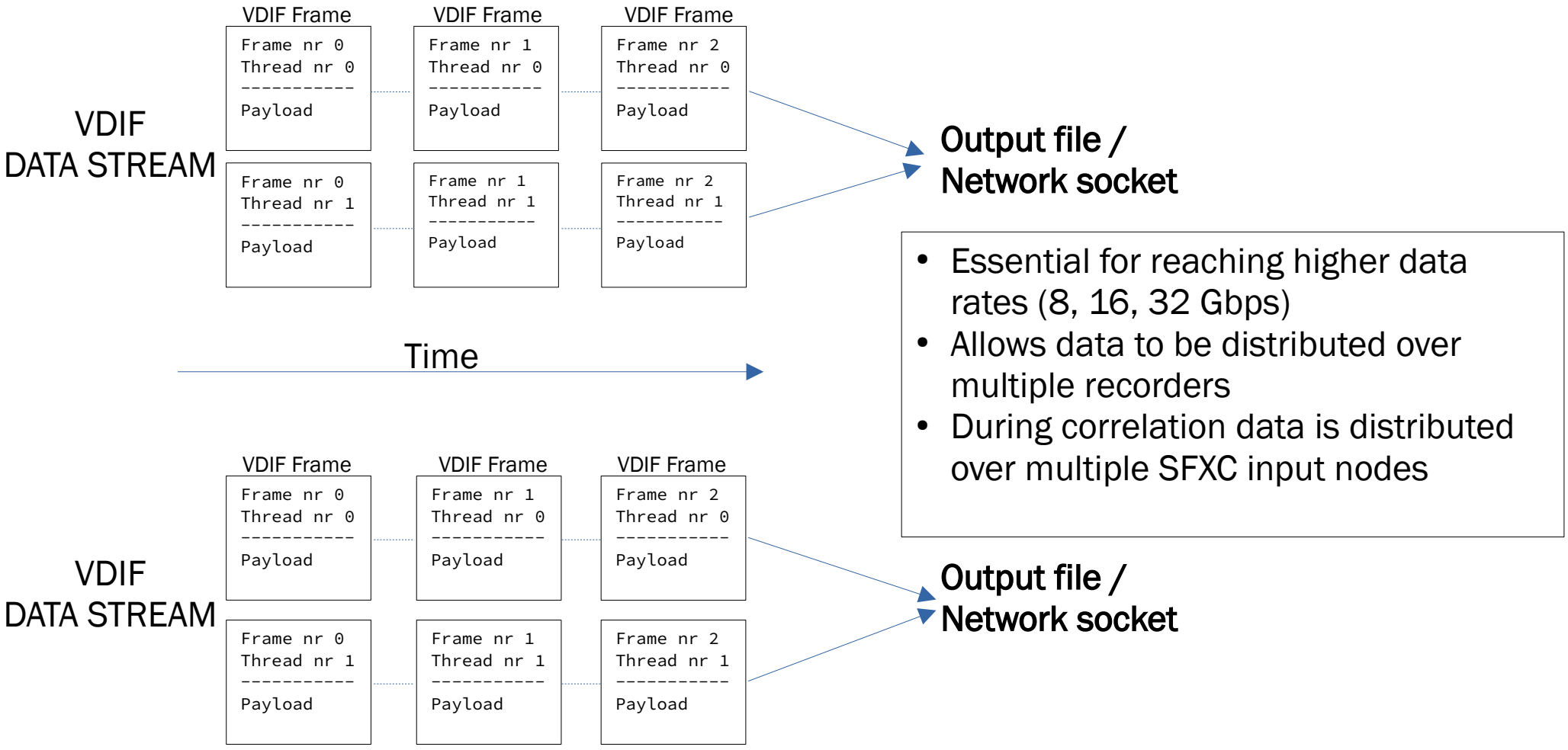
VDIF – Single data stream



VDIF – multiple data streams



VDIF – multiple data streams



- Essential for reaching higher data rates (8, 16, 32 Gbps)
- Allows data to be distributed over multiple recorders
- During correlation data is distributed over multiple SFXC input nodes

Tour of the VEX file: $THREADS

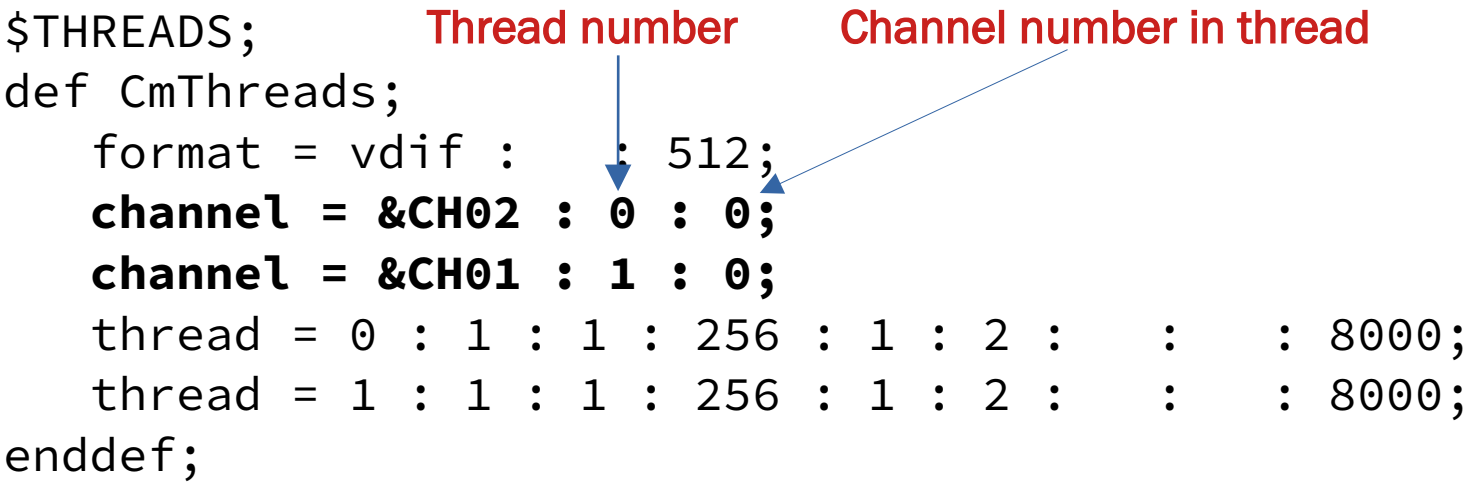
Configuration #1: Multiple threads, one channel per thread

```
$THREADS;
def CmThreads;
   format = vdif :   : 512;
   channel = &CH02 : 0 : 0;
   channel = &CH01 : 1 : 0;
   thread = 0 : 1 : 1 : 256 : 1 : 2 :   :   : 8000;
   thread = 1 : 1 : 1 : 256 : 1 : 2 :   :   : 8000;
enddef;
```

- Total data rate is 512 Mbps

Tour of the VEX file: $THREADS

Configuration #1: Multiple threads, one channel per thread



- Total data rate is 512 Mbps
- Channel labels as defined in the **$FREQ** section

Tour of the VEX file: $THREADS

Configuration #1: Multiple threads, one channel per thread

```
$THREADS;
def CmThreads;
   format = vdif :   : 512;
   channel = &CH02 : 0 : 0;
   channel = &CH01 : 1 : 0;
   thread = 0 : 1 : 1 : 256 : 1 : 2 :   :   : 8000;
   thread = 1 : 1 : 1 : 256 : 1 : 2 :   :   : 8000;
enddef;
```

Bits per sample

Frame size

Thread number

Ignored

Data rate of thread

Number of channels in thread

Tour of the VEX file: $THREADS

Configuration #2: One thread, multiple channels per thread

```
def EfThreads;
   format = vdif :   : 1024;
   channel = &CH01 : 0 : 4;
   channel = &CH02 : 0 : 6;
   channel = &CH03 : 0 : 0;
   channel = &CH04 : 0 : 2;
   channel = &CH05 : 0 : 5;
   channel = &CH06 : 0 : 7;
   channel = &CH07 : 0 : 1;
   channel = &CH08 : 0 : 3;
   thread = 0 : 1 : 1 : 1024 : 8 : 2 :   :   : 8000;
enddef;
```

Channel indices are in a different order than the channel labels

Single thread

8 Channels

Tour of the VEX file: $DATASTREAMS

Two datastreams, one thread per datastream

```
def Ef-8000-DS;
   datastream = &DS1 : VDIF : DS1;
   datastream = &DS2 : VDIF : DS2;
   thread = &DS1 : &thread0 : 0 : 4 : 64 Ms/sec : 2 : real : 8000;
   thread = &DS2 : &thread1 : 0 : 4 : 64 Ms/sec : 2 : real : 8000;
   channel = &DS1 : &thread0 : &CH01 : 2;
   channel = &DS1 : &thread0 : &CH02 : 3;
   channel = &DS1 : &thread0 : &CH03 : 0;
   channel = &DS1 : &thread0 : &CH04 : 1;
   channel = &DS2 : &thread1 : &CH05 : 2;
   channel = &DS2 : &thread1 : &CH06 : 3;
   channel = &DS2 : &thread1 : &CH07 : 0;
   channel = &DS2 : &thread1 : &CH08 : 1;
Enddef;
```

Number channels

Bits per sample

Sample rate

Channel within thread

Datastream label

Tour of the VEX file: $CLOCK

```
$CLOCK;
def CM;
   clock_early = 2024y061d12h00m00s : -12.7725 usec : 2024y061d13h30m00s : 3.6300000e-8 usec/sec;
enddef;
def DA;
   clock_early = 2024y061d12h00m00s : -12.8055 usec : 2024y061d13h30m00s : 3.6300000e-8 usec/sec;
enddef;
```

Valid from Clock offset Epoch Clock rate

- Clock offsets and rates
- Usually initial values from GPS will be available
- Ultimately will need to be measured using the correlator (see tutorial)

Tour of the VEX file: $EOP

```
def EOP060;
   TAI-UTC = 37 sec;
   eop_ref_epoch = 2024y060d00h00m00s;
   eop_interval = 24 hr;
   num_eop_points = 3;
   num_nut_points = 3;
   delta_psi = -0.109842 asec : -0.109976 asec : -0.110136 asec;
   delta_eps = -0.007223 asec : -0.007486 asec : -0.007893 asec;
   x_wobble = 0.007865 asec : 0.005660 asec : 0.004414 asec;
   y_wobble = 0.267828 asec : 0.269789 asec : 0.272392 asec;
   ut1-utc = -0.0031534 sec : -0.0033589 sec : -0.0034879 sec;
enddef;
```

- Earth orientation parameters (EOP)
- Needed for computing the delay model
- EOPs are continuously updated, significant changes in the first week after observation
  Link : ftps://gdc.cddis.eosdis.nasa.gov/vlbi/gsfc/ancillary/solve_apriori/usno_finals.erp

Preparing correlation VEX file

- VEX files created using scheduling software, e.g. (py)sched, lack a number of sections:
  - $THREADS / $BITSTREAMS
  - $EOP
  - $CLOCK
- Utility `prepare_vex.py`
  - Downloads EOP from NASA GDC, and creates $EOP
  - Creates $CLOCK with all clocks zero
  - Creates $THREADS (not $BITSTREAMS or $DATASTREAMS) using heuristics based on the station name.
- See first tutorial session

Troubleshooting: vdif_print_headers.py

```
vdif_print_headers -n 1 /data/n24l2/files/n24l2_ef_no0004

2024y144d12h41m47.000s ,frame_nr = 10422, thread_id = 0, nchan = 8,
invalid = 0, legacy = 0, station = NA, complex = 0, bps-1 = 1,
data_size = 8032
```

- Quick check of data file
  - Is number of channels as expected
  - Does data frame size match vex file (note above number include headers)
  - For multi-threaded VDIF do the `thread_ids` match the vex file

Troubleshooting: thread mapping

```
def EfThreads;
   format = vdif :   : 1024;
   channel = &CH01 : 0 : 2;
   channel = &CH02 : 0 : 2;
   channel = &CH03 : 0 : 2;
   channel = &CH04 : 0 : 2;
   channel = &CH05 : 0 : 2;
   channel = &CH06 : 0 : 2;
   channel = &CH07 : 0 : 2;
   channel = &CH08 : 0 : 2;
   thread = 0 : 1 : 1 : 1024 : 8 : 2 :   :   : 8000;
enddef;
```

- If thread mapping is unknown broadcast single channel
- Check which channel has a detection, and repeat

Correlator control files

- Describes how the correlator is run
  - Number of spectral channels
  - Integration times
  - Location of input data
  - Which scan(s), stations, sub-bands, … to correlate.
  - Etc…
- JSON format, human readable and easily parsable in e.g. python
- Basic control files can be generated using the generate_jobs.py (see tutorial)
- Documentation: https://jradcliffe5.github.io/sfxc_workshop_2025/Control_file_parameters.html

Example control file

```
{
    "setup_station": "Ef",
    "data_sources": {
        "Cm": [ "file:///data/n24l2_cm_no0004.vdif",
                "file:///data/n24l2_cm_no0005.vdif"],
        "Ef": [ "file:///data/n24l2_ef_no0004"]

    },
    "start": "2024y144d12h41m47s",
    "stop": "2024y144d12h47m13s",
    "stations": [ "Cm", "Ef"],
    "channels": [ "CH01", "CH02" ]
    "number_channels": 64,
    "integr_time": 2,
    "output_file": "file:///data/out/n24l2/n24l2_no0004.cor",
    … entries omitted …
}
```

Example control file

```
{
    "setup_station": "Ef",
    "data_sources": {
        "Cm": [ "file:///data/n24l2_cm_no0004.vdif",
                "file:///data/n24l2_cm_no0005.vdif"],
        "Ef": [ "file:///data/n24l2_ef_no0004"]

    },
    "start": "2024y144d12h41m47s",
    "stop": "2024y144d12h47m13s",
    "stations": [ "Cm", "Ef"],
    "channels": [ "CH01", "CH02" ]
    "number_channels": 64,
    "integr_time": 2,
    "output_file": "file:///data/out/n24l2/n24l2_no0004.cor",
    … entries omitted …
}
```

Start- and stop time of correlation

Integration time

Example control file

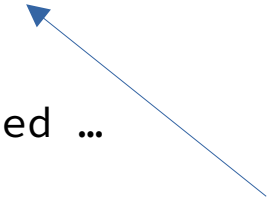
```
{
    "setup_station": "Ef",
    "data_sources": {
        "Cm": [ "file:///data/n24l2_cm_no0004.vdif",
                "file:///data/n24l2_cm_no0005.vdif"],
        "Ef": [ "file:///data/n24l2_ef_no0004"]

    },
    "start": "2024y144d12h41m47s",
    "stop": "2024y144d12h47m13s",
    "stations": [ "Cm", "Ef"],
    "channels": [ "CH01", "CH02" ]
    "number_channels": 64,
    "integr_time": 2,
    "output_file": "file:///data/out/n24l2/n24l2_no0004.cor",
    … entries omitted …
}
```

Location of data

Which stations to correlate

data_sources with datastreams

```
{
    "setup_station": "Ef",
    "data_sources": {
        "Cm": { "DS0": [ "file:///data/n24l2_cm_no0004.vdif",
                         "file:///data/n24l2_cm_no0005.vdif"]},
        "Ef": { "DS0": [ "file:///data/n24l2_ef_no0004_ds0"],
                "DS1": [ "file:///data/n24l2_ef_no0004_ds1"]}
    },

    … entries omitted …
}
```



Datastream labels as defined in the $DATASTREAMS section

Example control file

```
{
    "setup_station": "Ef",
    "data_sources": {
        "Cm": [ "file:///data/n24l2_cm_no0004.vdif",
                "file:///data/n24l2_cm_no0005.vdif"],
        "Ef": [ "file:///data/n24l2_ef_no0004"]

    },
    "start": "2024y144d12h41m47s",
    "stop": "2024y144d12h47m13s",
    "stations": [ "Cm", "Ef"],
    "channels": [ "CH01", "CH02" ]
    "number_channels": 64,
    "integr_time": 2,
    "output_file": "file:///data/out/n24l2/n24l2_no0004.cor",
    … entries omitted …
}
```

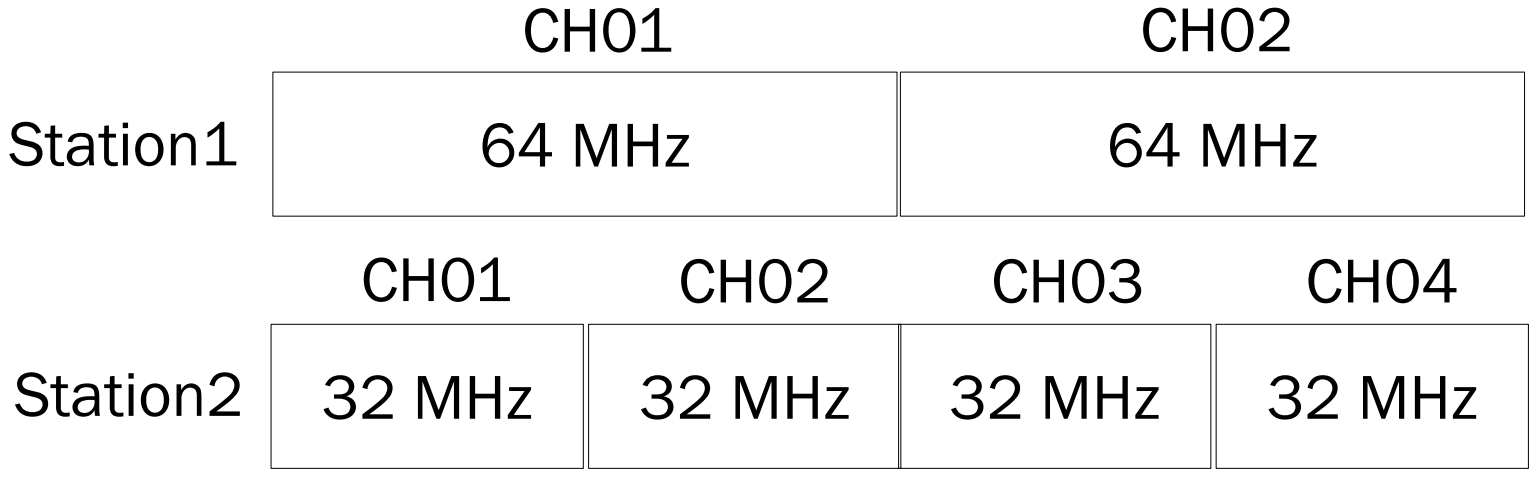
Use the channels as defined for this stations

Which channels to correlate
(as defined by setup_station)

Number of spectral points to output

Mixed bandwidth

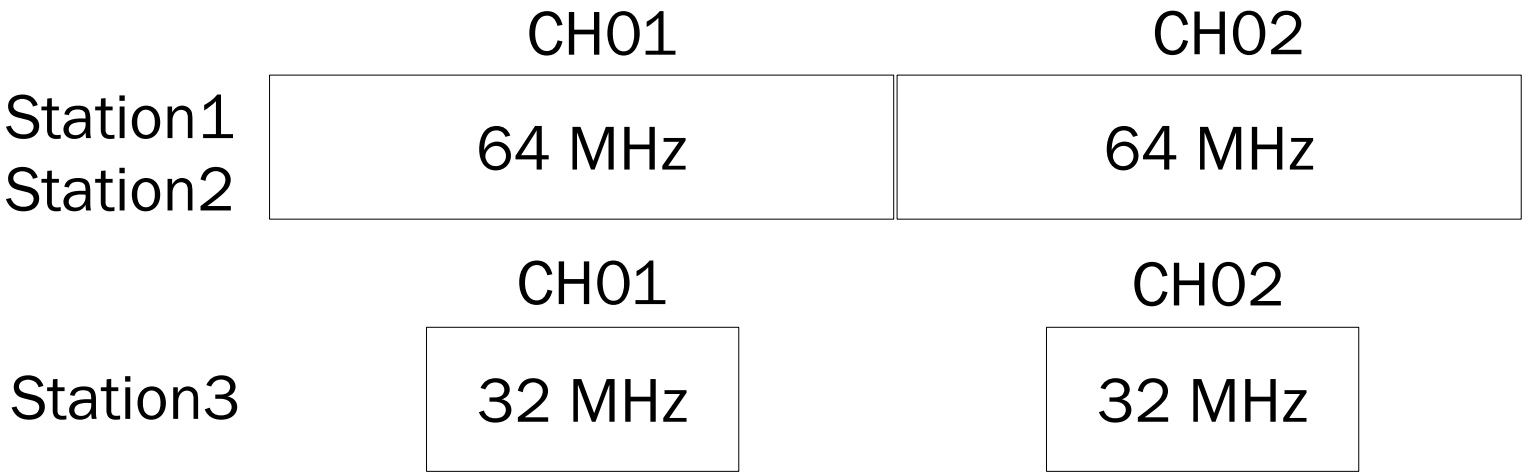
Multiple 32 MHz channels complete overlapping with 64 MHz channels



- SFXC can split wider bands into multiple smaller bands
- But it cannot combine multiple smaller bands into a larger band
- Choose Station2 as setup station

Mixed bandwidth

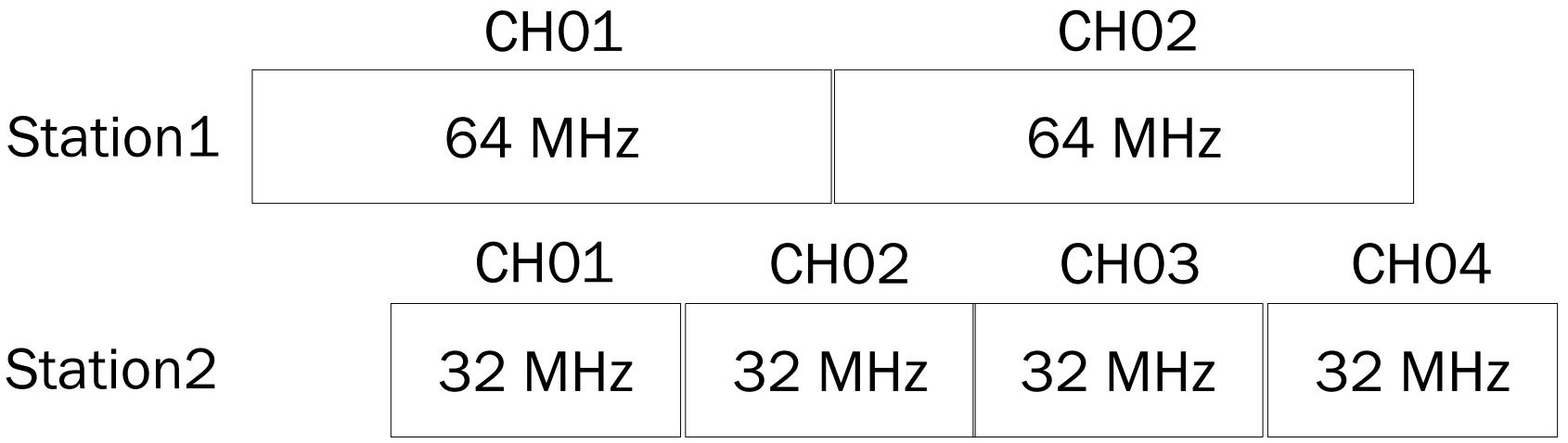
Two stations with 64 MHz channels and a station with two 32 MHz channels



- When Station1 is setup station
  - Station1 – Station2: 64MHz bands
  - Station1/2 - Station3: 64 MHz bands, 50% of band is zeros
- When Station3 is setup station
  - Station1 – Station2: 32MHz bands, matching the channels from Station3
  - Station1/2 - Station3: 32 MHz bands

Mixed bandwidth

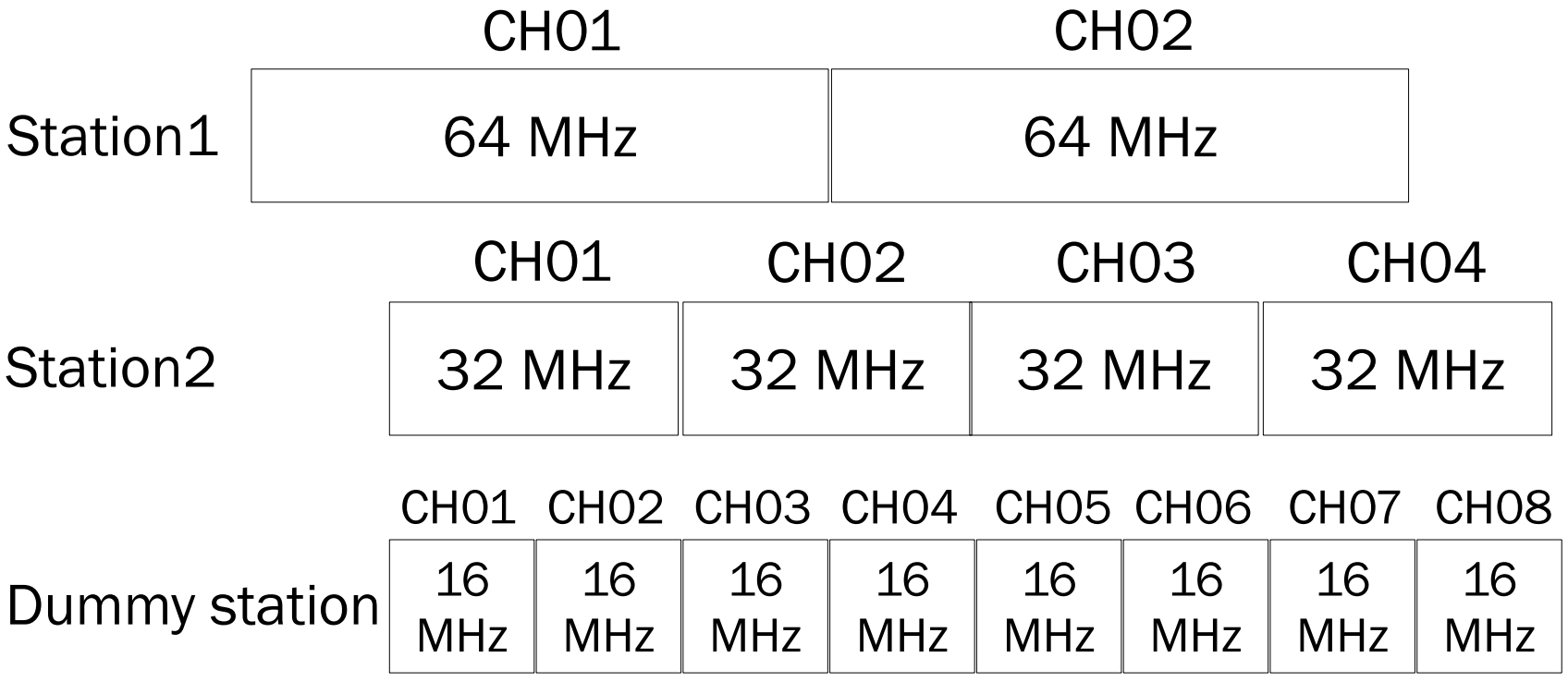
64 MHz channels with half overlapping 32 MHz channels



- How to correlate CH02, and CH04 from Station2?
  - SFXC can't combine bands to correlate CH02
  - There is no data for the upper 16 MHz of CH04, SFXC will not correlate this channel

Mixed bandwidth

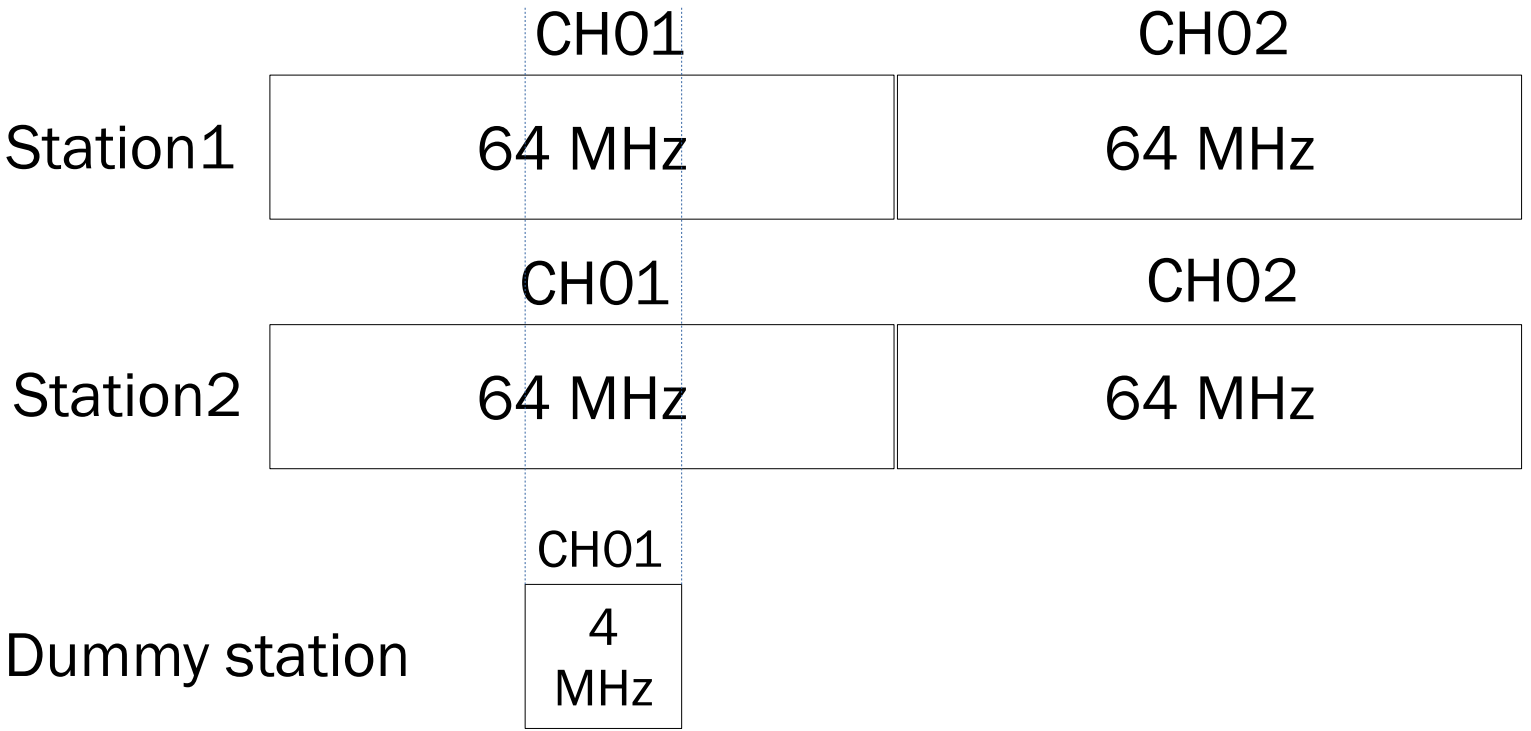
64 MHz channels with half overlapping 32 MHz channels



Solution: Add dummy station to VEX file with above $FREQ section and use that as setup station (see tutorial)

Zoom bands

64 MHz channels, but only 4 MHz is of interest due to spectral line



Similarly dummy stations can also be used to create a zoom band