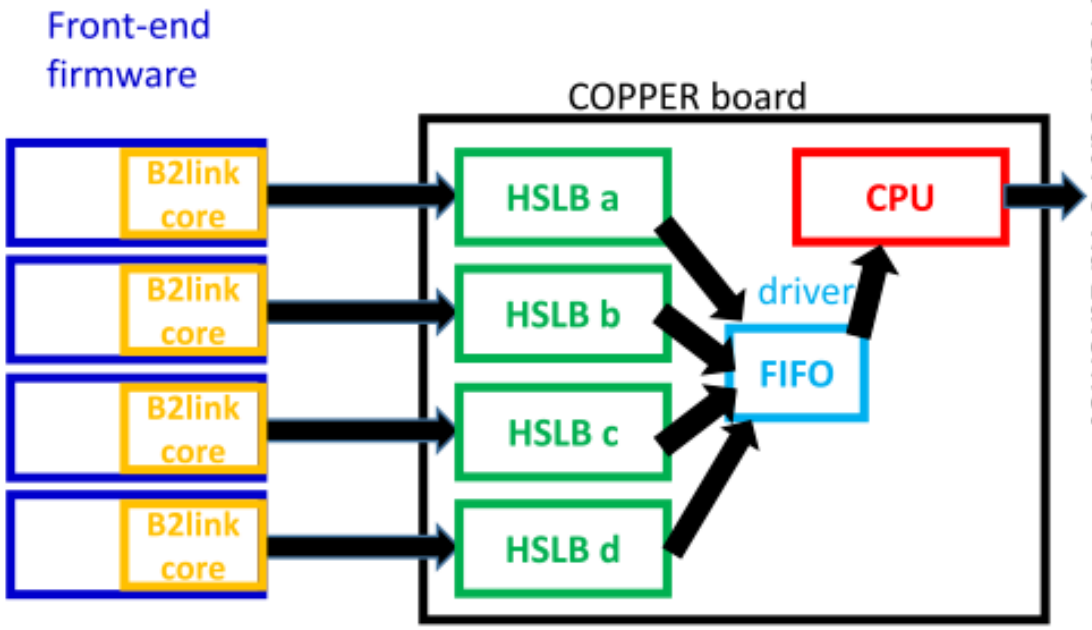
RawCOPPER data format

Aug. 23, 2014 (svn rev. 12453)

Satoru Yamada

1, Overview of RawCOPPER format (one data block from a COPPER board)

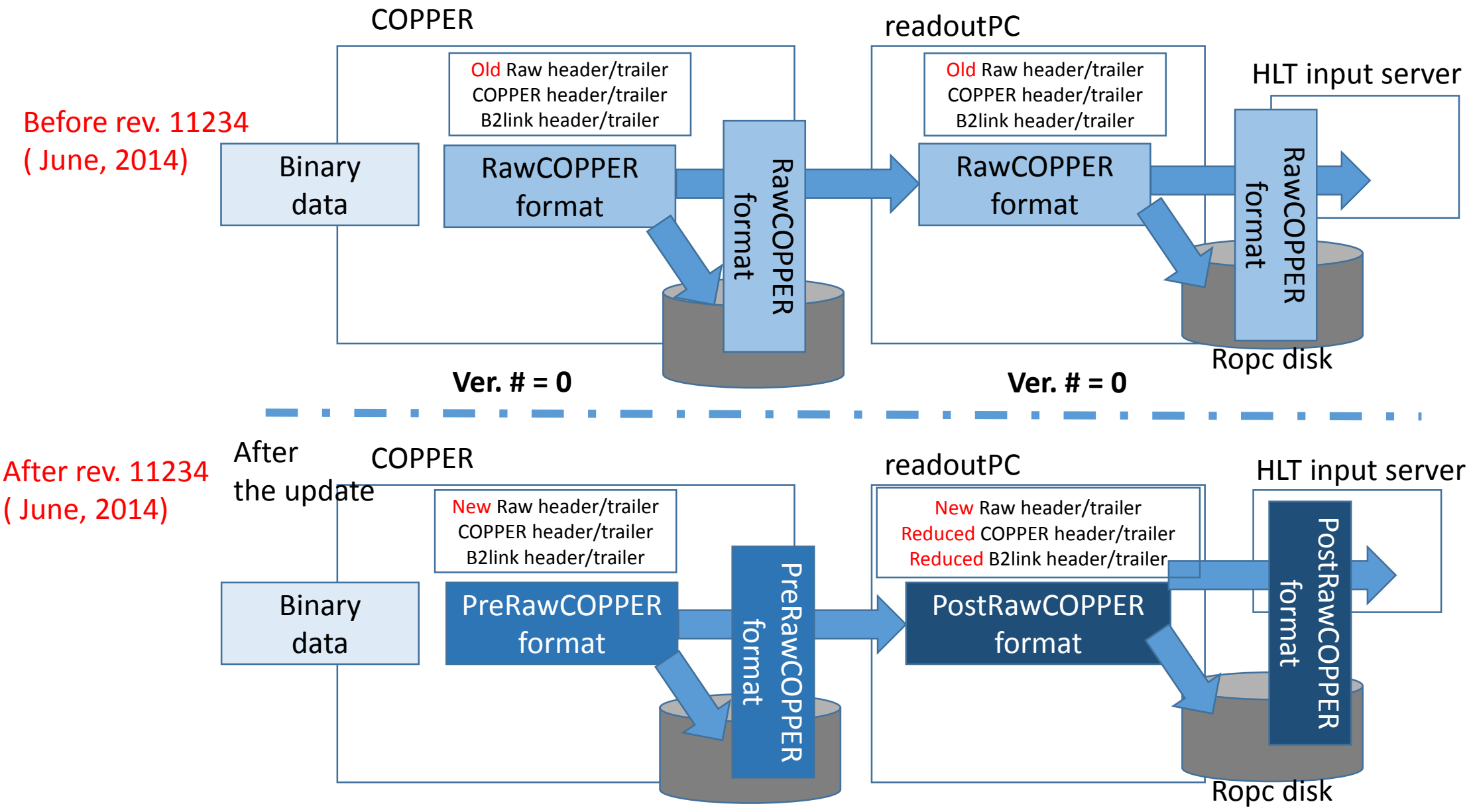
RawCOPPER header/trailer -> See Sec. 2
COPPER header/trailer -> See Sec.3
B2link(FEE+HSLB) header/trailer -> See Sec.4
Detector buffer -> Untouched by DAQ



- RawCOPPER header
  - COPPER header
    - B2link HSLB header (slot A FINNESSE)
      - B2link FEE header(slot A FINNESSE)
        - Data contents(Detector buffer) (slot A FINNESSE)
      - B2link FEE trailer (slot A FINNESSE)
    - B2link HSLB trailer (slot A FINNESSE)
    - B2link HSLB header (slot B FINNESSE)
      - B2link FEE header(slot B FINNESSE)
        - Data contents(Detector buffer) (slot B FINNESSE)
      - B2link FEE trailer (slot B FINNESSE)
    - B2link HSLB trailer (slot B FINNESSE)
    - B2link HSLB header (slot C FINNESSE)
      - B2link FEE header(slot C FINNESSE)
        - Data contents(Detector buffer) (slot C FINNESSE)
      - B2link FEE trailer (slot C FINNESSE)
    - B2link HSLB trailer (slot C FINNESSE)
    - B2link HSLB header (slot D FINNESSE)
      - B2link FEE header(slot D FINNESSE)
        - Data contents(Detector buffer) (slot D FINNESSE)
      - B2link FEE trailer (slot D FINNESSE)
    - B2link HSLB trailer (slot D FINNESSE)
  - COPPER trailer
- RawCOPPER trailer

HSLB : High speed link board

1-1, Online header/trailer reduction

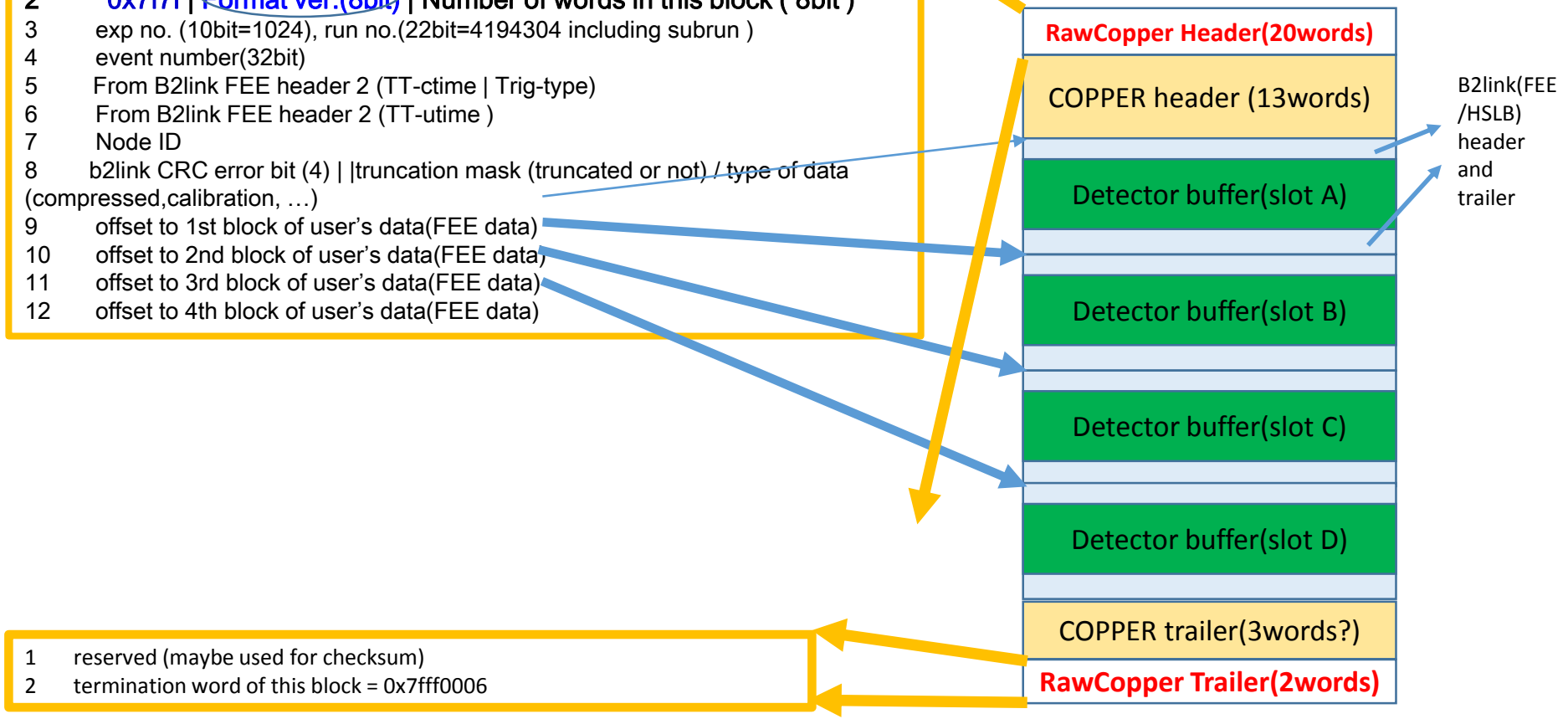


- PreRawCOPPER format
    - If you store data by COPPER CPU, then output data will be in Pre(reduction)RawCOPPER format.
- PostRawCOPPER format
    - Store the data downstream from readout PC, the output data will be in Post(reduction)RawCOPPERFormat

2-1, “RawCOPPER header/trailer” format in PreRawCOPPER format (ver. 1+0x80)

Use this version number to distinguish
Different data format.
Ver.0 : to 2014. June(including DESY test)
ver.1 : from June.2014

1 Number of total words
2 0x7f7f | Format ver.(8bit) | Number of words in this block (8bit)
3 exp no. (10bit=1024), run no.(22bit=4194304 including subrun)
4 event number(32bit)
5 From B2link FEE header 2 (TT-ctime | Trig-type)
6 From B2link FEE header 2 (TT-utime)
7 Node ID
8 b2link CRC error bit (4) | |truncation mask (truncated or not) / type of data (compressed,calibration, …)
9 offset to 1st block of user’s data(FEE data)
10 offset to 2nd block of user’s data(FEE data)
11 offset to 3rd block of user’s data(FEE data)
12 offset to 4th block of user’s data(FEE data)



1 reserved (maybe used for checksum)
2 termination word of this block = 0x7fff0006

2-2, “RawCOPPER header” and trailer format in PostRawCOPPER format (ver.0x01)

Same as PreRawCOPPER format

2-3, tentative format of 32bit node ID (A.K.A. subsystem ID)

Format :

(31-24) Detector ID : 8bit=256 : detector ID
(9-0) lower bits of COPPER ID : 10bit (1024)

Detector ID :

Detector ID (Defined in rawdata/dataobjects/include/RawCOPPERFormat.h)

- #define SVD_ID 0x01000000 // tentative
- #define CDC_ID 0x02000000 // tentative
- #define BPID_ID 0x03000000 // tentative
- #define EPID_ID 0x04000000 // tentative
- #define BECL_ID 0x05000000 // tentative
- #define EECL_ID 0x06000000 // tentative
- #define BKLM_ID 0x07000000 // tentative
- #define EKLM_ID 0x08000000 // tentative

Full COPPER ID :

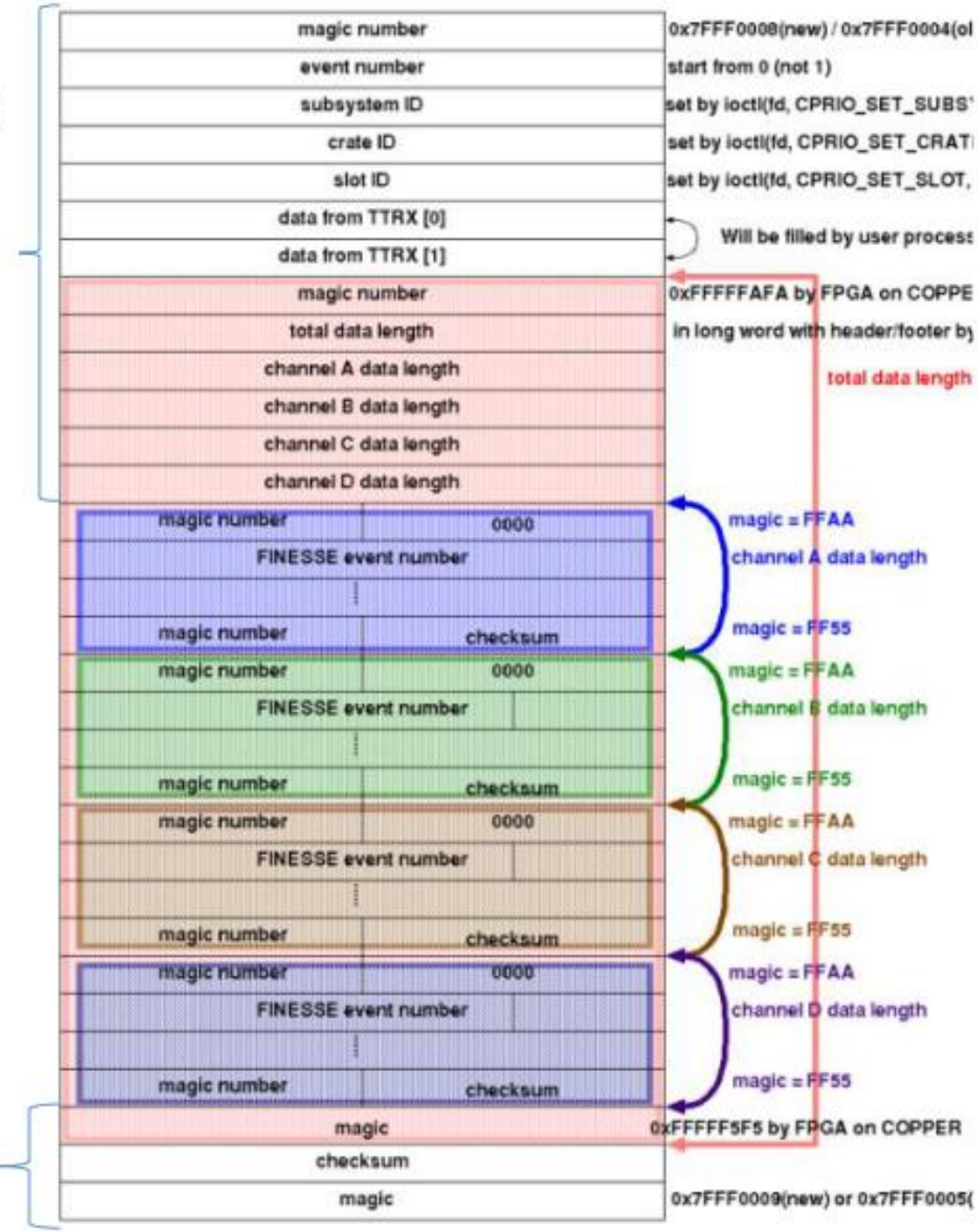
Full COPPER ID can be reconstructed by “(Detector ID >> 24) * 1000 + COPEPR ID(12bit) “
e.g. NodeID = 0x0600000a -> COPPER ID = cpr6010
NodeID = 0x0100000a -> COPPER ID = cpr1010
A label of COPPER ID will be attached on the front of a COPPER board

Node ID = “TTD ” = 0x54544420 and “FTSW” are reserved for VME CPU and FTSW now.

3-1, COPPER header and trailer in PreRawCOPPER format (ver. 1 + 0x80)

COPPER header

COPPER Trailer



3-2, COPPER header and trailer in PostRawCOPPER format (ver. 0x01)

No COPEPR header and trailer in Post reduction rawcopper format.

4-1, B2link FEE header/Trailer, B2link HSLB header/Trailer in PreRawCOPPERFormat (ver. 0x01 + 0x80)

From Nakao-san's Belle2link User guide (June 10, 2014):
You can download from 18 th B2GM indico page
http://kds.kek.jp/getFile.py/access?contribId=132&sessionId=28&resId=0&materialId=0&confId=15329

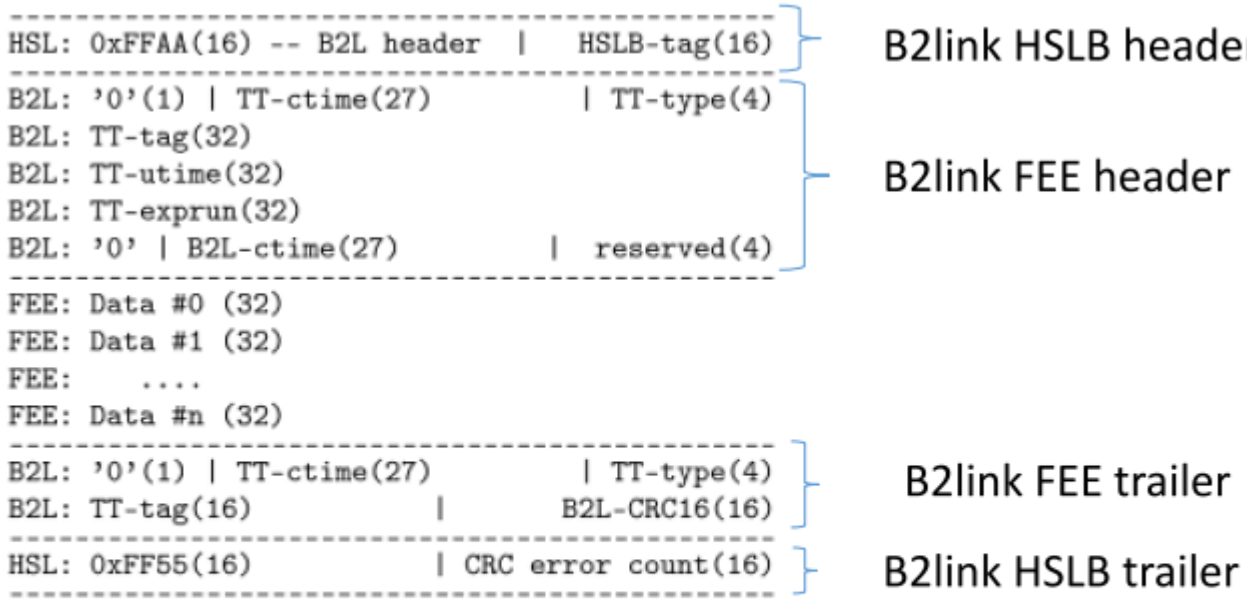


Figure 5: Data format as read out by the COPPER. The header and trailer words labelled with HSL are attached by HSLB, the words with B2L are attached by the belle2link component, and the words with FEE are those written into the belle2link component by the frontend firmware.

NOTICE :
To produce this format, the b2tt core used in
the FEE firmware should be the latest.
Please see Nakao-san's following e-mails :
[b2link_ml:0143] Belle2link version 0.01 - SVN update
And
[b2link_ml:0144] Re: Belle2link version 0.01 - SVN update .

4-2, B2link FEE header/Trailer, B2link HSLB header/Trailer in PostRawCOPPERFormat (ver. 0x01)

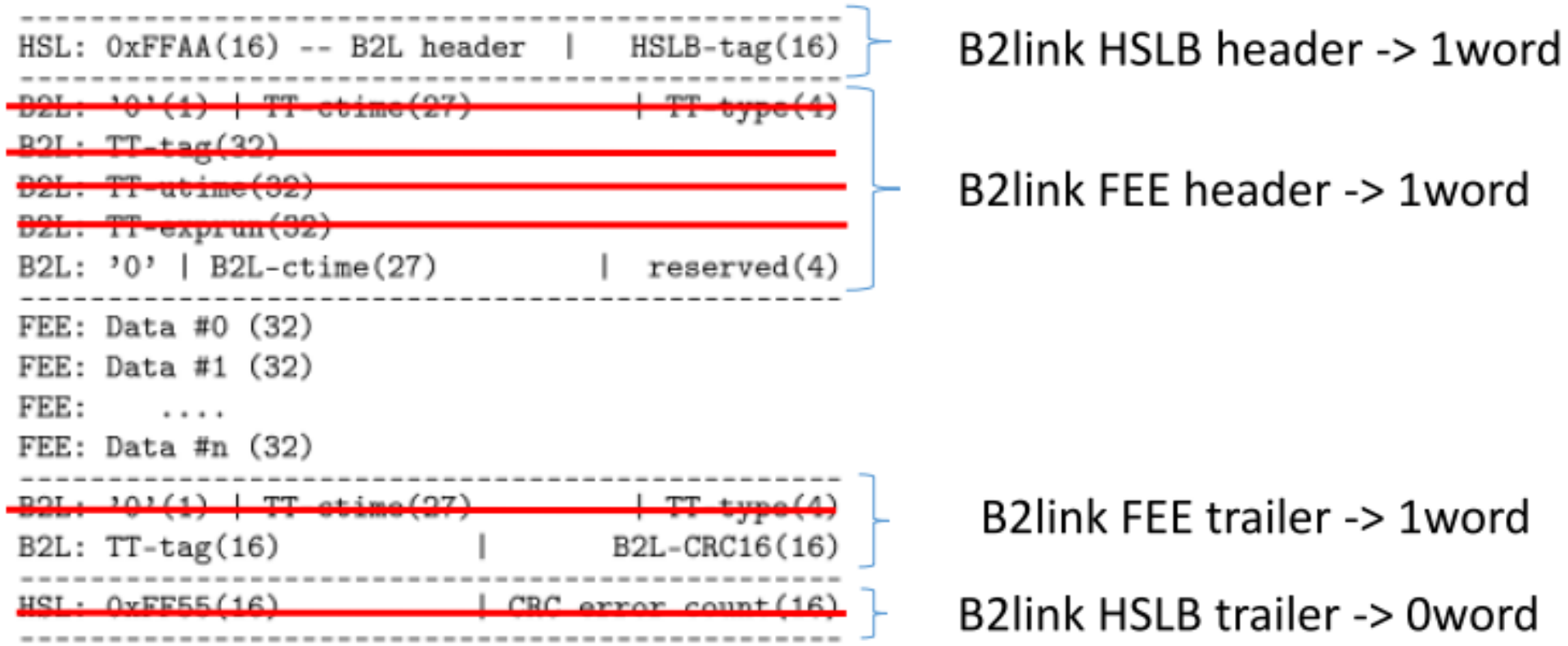


Figure 5: Data format as read out by the COPPER. The header and trailer words labelled with **HSL** *are attached by HSLB, the words with* **B2L** *are attached by the belle2link component, and the words with FEE are those written into the belle2link component by the frontend firmware.*

4-3, Older B2link header/trailer formats

At DESY test in January of 2014

From Nakao-san's B2GM slides:
http://kds.kek.jp/getFile.py/access?contribId=143&sessionId=38&resId=0&materialId=slides&confId=13911

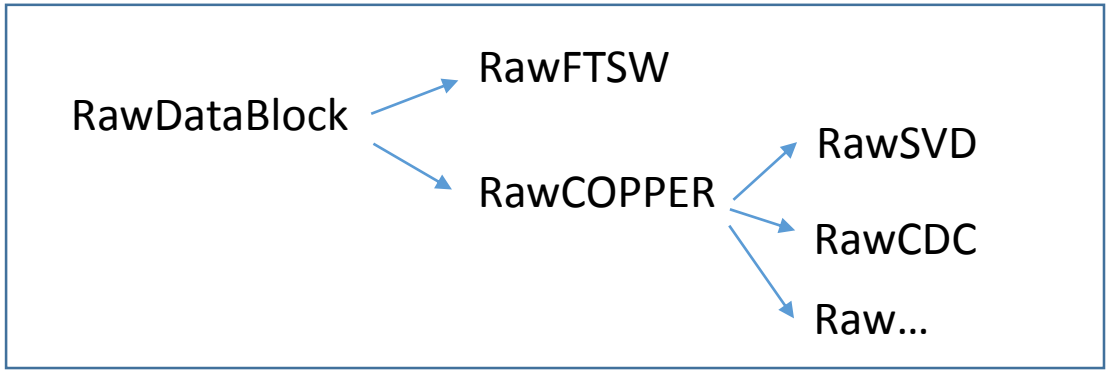
Data format (Final?)

The format used at the telescope test

```
-------------------------------------------------
HSL: 0xFFAA(16) --- B2L header  |      HSLB-tag(16)
-------------------------------------------------
B2L: '0'(1) | TT-ctime(27)              | TT-type(4)
B2L: TT-tag(32)
B2L: TT-utime(32)
B2L: TT-exprun(32)
B2L: '0' | B2L-ctime(27)        |     debug-flag(4)
-------------------------------------------------
FEE: Data #0 (32)
FEE: Data #1 (32)
FEE:    ....
FEE: Data #n (32)
-------------------------------------------------
B2L: TT-tag(16)                 |  B2L-checksum(16)
-------------------------------------------------
HSL: 0xFF55(16)                 | HSLB checksum(16)
-------------------------------------------------
```

- tag (event number) and utime to be increased to 32-bit (done), HSLB-checksum, B2L-checksum to be added

5-1, RawDataBlock object (to handle Raw data from COPPER board)



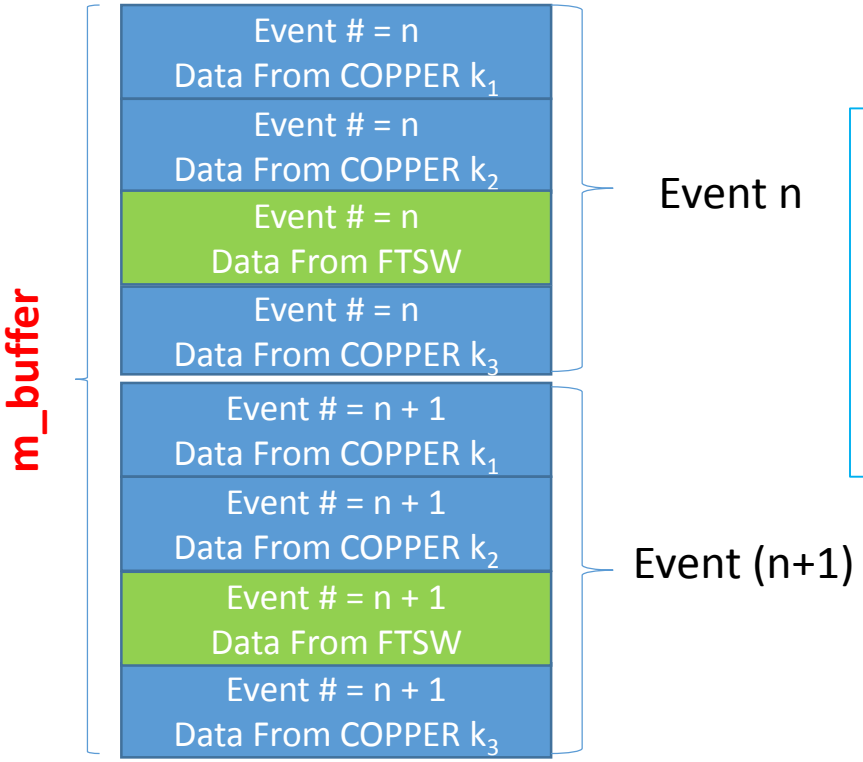
Source code :

https://belle2.cc.kek.jp/svn/trunk/software/rawdata/dataobjects/

```
RawDataBlock{
        methods to access data;
        int m_num_nodes; // # of nodes
        int m_num_events;// # of events

        int* m_buffer;  -> buffer for data
}
```

Example of data structure

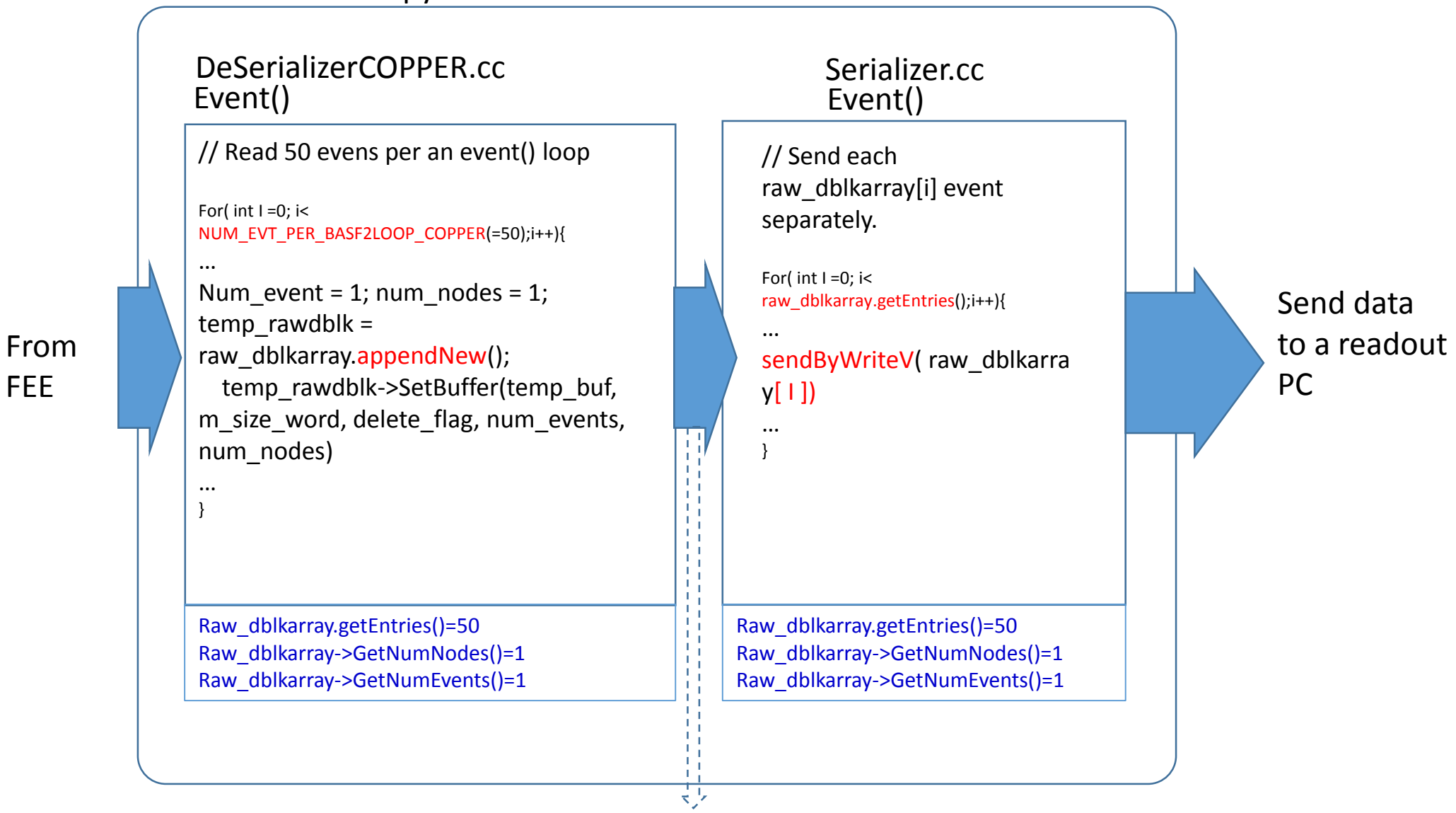


In this example,
M_num_nodes = 4
M_num_events = 2.

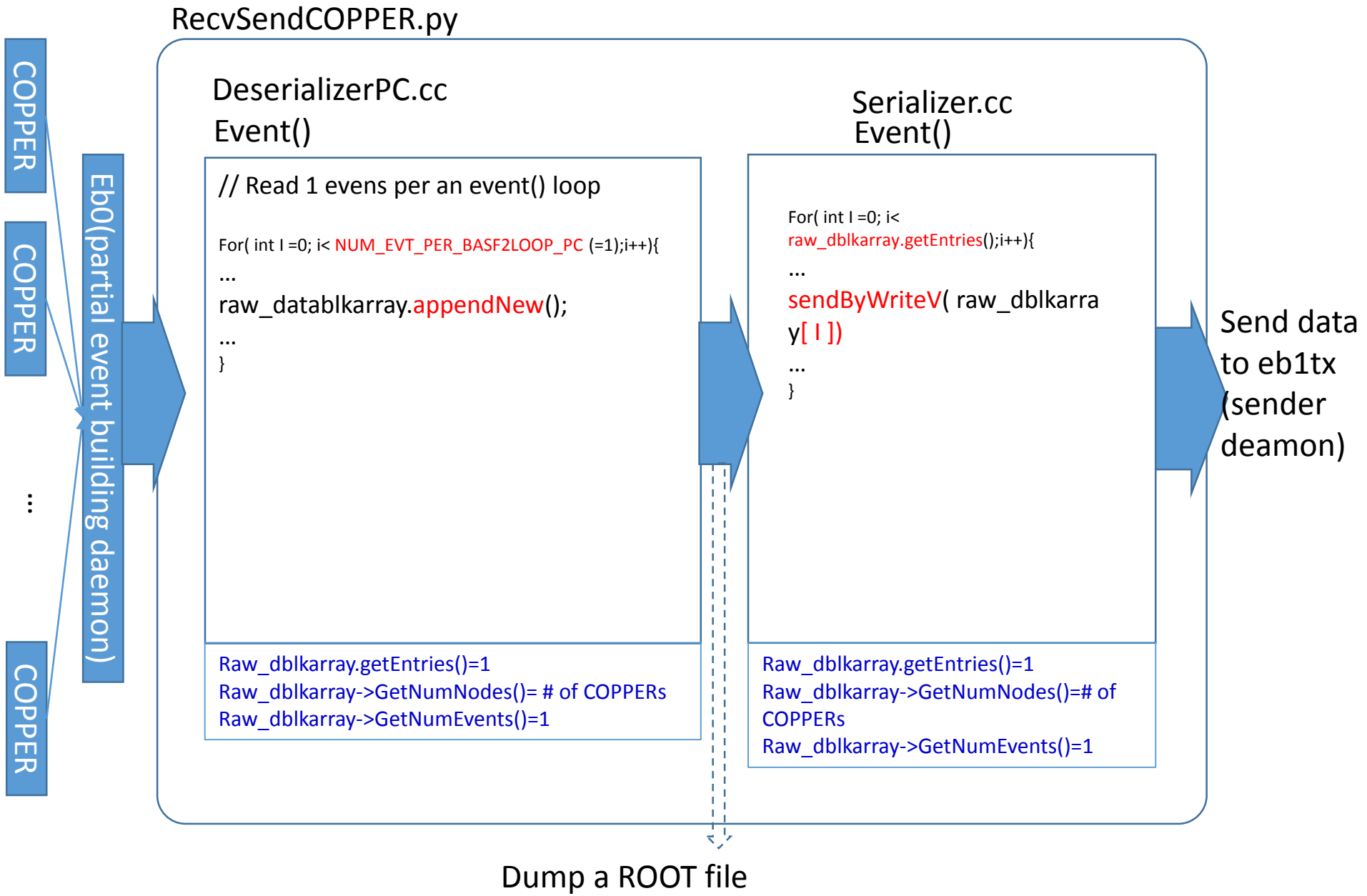
# of data blocks = 4 * 2 = 8

Example of Data handling on COPPER (as of rev.12453)

RecvSendCOPPER.py



Example of Data handling on a readout PC (as of rev.12453)



Example :
# of event and node in one RawDataBock object :
Output by RecvStream1.py and 3 RecvSendCOPPER.py processes

RecvStream1.py on a readout PC
event k : Ser len 135 numeve 1 node 3
event k : Des len 132 numeve 1 node 3
event k+1 : Ser len 132 numeve 1 node 3
event k+1 : Des len 129 numeve 1 node 3

RecvSendCOPPER.py on COPPER1
event l : Ser len 66 numeve 1 node 1
event l+1:Ser len 84 numeve 1 node 1

RecvSendCOPPER.py on COPPER2
event m : Ser len 78 numeve 1 node 1
event m+1 : Ser len 77 numeve 1 node 1

RecvSendCOPPER.py on COPPER3
event n : Ser len 84 numeve 1 node 1
event n+1 : Ser len 64 numeve 1 node 1

2-2, Rawdata Unpacker for new and old data formats

Data taken at the DESY beam test(old format) can be read with the latest rawdata package
-> by checking data ver. In header.

New RawCOPPER class

- No change in style of the member functions -> No effect on derived class
- Does not have a format information in itself
  - Format class contains format information
    - RawCOPPERformat.cc -> the latest format
      - RawHeader.cc
    - RawCOPPERformat_v0.cc -> an old format
      - RawHeader_v0.cc
  - Assgin a format class to m_access in CheckVersionSetBuffer()
  - Use m_access to access buffer contents

```
inline int RawCOPPER::GetExpNo(int n)
{
  CheckVersionSetBuffer();
  return m_access->GetExpNo(n);
}

inline int RawCOPPER::GetRunNo(int n)
{
  CheckVersionSetBuffer();
  return m_access->GetRunNo(n);
}
```

Notice :
- RawCOPPER class supports both formats for a while (0.5-1 year after the format becomes stable?).
- In that case, the latest RawCOPPER class cannot be used to read old format
- Of course, you can use old rawdata repository to read old format
- For ver.0 format, use rawdata repository before 11228

Revision History of this document

- Jan.5, 2014 rev. 8376 : Add definition of tentative subsysID format
- Dec. 16, 2013 rev.7974 :
    - Add B2linkFEE header format
    - Add comments about handling StoreArray when unpacking Raw*** data.
- Oct.21, 2013 :rev.7133
    - Add instruction about Rawdata unpacking program
- Oct. 18, 2013 :rev. 7095
    - 1 st draft
- Jun. 23, 2014 : rev. 11234
    - Online (header/trailer) reduction scheme on readout PC is introduced
        - RawHeader format is changed
        - COPPER header/trailer format is changed
    - Nakao-san updated B2LFEE/HSLB header/trailer format
        - See [b2link_ml:0144] Re: Belle2link version 0.01 - SVN update
- Aug. 23, 2014: rev. 12453
    - Add a description of how RawDataBLock objects are handled by the actual DAQ program.