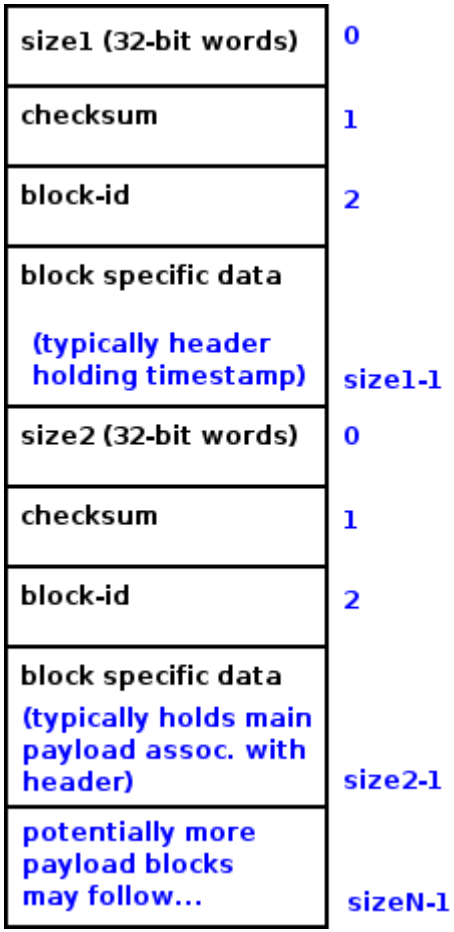
Adapting MINOS Online/Offline code to Daya Bay

- Overview of proposal
- Description of the format and code
- Decisions needed

Overview of Proposal

- Need a raw data format and a mechanism to bring raw data from online to offline software contexts:
- Proposal: steal from MINOS!
  - Simple flat online raw data blocks/records
  - RotoTalk network protocol
  - “rotorooter” server
  - RawData classes wrap unmolested raw blocks and provide unpacking methods
- Already have permission from author (Robert Hatcher, FNAL).

Online Raw Data Format



- Raw data record made of multiple raw data blocks
- Simple array of integers
- Each block:
  - Header:
    - size
    - checksum
    - block ID
  - Block data, layout determined by block ID
- Initial block: header with timestamp, summary data
- Total size required when sent

Notes on Raw Data Blocks

- All data is integer
  - byte ordering handled, but
  - a float must be “equivalenced” to an int, can cause trouble with mixed FP standards
- Online is free to choose data block layout
- Block ID encodes:
  - data vs. MC
  - data source (detector)
  - major version: block type
  - minor version: layout version

Must strictly maintain versioning!

RotoTalk protocol

- Simple C implementation, can be used from any language (eg. Python, C++)
- Connects client to “rotorooter” server
- Provides communication methods:
  - configuration options
  - connect, shutdown
  - open, close file
  - send data record (one or more blocks)

Rotorooter server

- Listens on port for RotoTalk connections
- Port number convention – one per stream
- Accepts the raw data records
- Creates associated RawXXXBlock subclass based on the block ID
- Saves RawDataBlock object to file
- Files open/close based on client RotoTalk commands, named by run/subrun or time

RawData classes

- General RawDataBlock holds raw data blocks, unmolested
- Specific RawXXXBlock subclass knows how to unpack raw data of type XXX
- Can organize RawData + RawHeader into full featured MINOS I/O streams, but may be more than we want.
- RawHeader holds timestamp, first raw block expected to provide this.

Decisions Needed

- Decide how much of the code we want
  - Just RawDataBlock, RotoTalk and rotorooter
  - Also MINOS I/O format (not simple Ttree) ?
  - Also MINOS persistency (provides automatic time syncronizing of independent streams) ?
- If just keep “R., R. and r” how to organize RawDataBlocks on disk?