## 3.1 Message Format and Field Encoding

Messages are composed by a header and a body where applicable.

Header: contains control information

Body: contains file chunk data

### Header

The header consists of a sequence of ASCII lines terminated by '0xD''0xA' (CRLF). Each header line is a sequence of fields separated by spaces. To note:

* there can be >1 space between header fields
* there may be 0 or more spaces after the last field in a line
* header terminates with an empty header line <CRLF><CRLF>

**Generic header format:**

<MessageType> <Version> <SenderID> <FileID> <ChunkNo> <ReplicationDeg> <CRLF>

Some fields do not appear on certain messages, but the fields that appear must follow the order above.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Length (bytes) | Format | Example |
| MessageType | Variable | ASCII specifying the action to be performed | PUTCHUNK, STORED |
| Version | 3 | “<n>.<m>” where <n><m> are digits | “1.0” for the base version |
| SenderID | Variable | ASCII representing integer ID of peer that sent the message | “2” |
| FileID | 64 | SHA256 hash of the filename that is encoded as Hex -> ASCII, which means that a hex value (0xB2) is sent as "B2"/"b2". Follows big endian order, from MSB to LSB | Hash is calculated as 0xAB10, string is “AB10” |
| ChunkNo | <=6 | ASCII representation of integer starting at 0, most significant digit first | 50 is “50” |
| ReplicationDeg | 1 | ASCII representation of integer up to 9 | “2” |

### Body

The body contains the data of a file chunk. The length is variable and can be up to 64KB (K=1000), if it’s smaller than 64KB then it’s the last chunk in a file. The body contents are interpreted as a byte sequence, no encoding should be done on it.