V5 (provisional)

1	2	3
4	5	6
7	8	9

| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte

Maximum size: 1000 bytes

Useful Size: 6 bytes

All integer values are unsigned

Field descriptions

- First byte: Version (5)
- Second byte: Position
- Third byte:
    - 2 == General error
    - 1 == Game complete
    - 0 == Game in progress
- Fourth byte: Modifier of the third byte
    - Case: General Error
        - 1 == Out of resources (can't accept a new game at this time)
        - 2 == Malformed/invalid request
        - 3 == Server shutdown
        - 4 == Client game timeout (server to client)
        - 5 == Try again
    - Case: Game complete

            - 1 == Draw
            - 2 == I win (sender of the message)
            - 3 == You win (receiver of the message)
        - Case: Game in progress
            - No Info Flag
    - Fifth Byte: Command
        - 0 == new game
        - 1 == move
    - Sixth Byte: Game Number, indicates to server what game is being played
    - All undefined bytes reserved for future use, can be considered as junk

Initial Handshake:

- To start a game there must be a handshake procedure:
    1. Client sends command “new game”, version number as first byte, other fields irrelevant
    2. Server Responds with game number, which will be used by both parties to identify future move, or general error, with error field set appropriately
    3. Game board is still blank, after receiving game number client makes first move

Normal Play:

- Run after initial handshake:
    1. Client sends move to server with command field set to move (1), and game number set to the clients game number, with end game fields set appropriately
    2. Server responds with move, sets end game field appropriately
- If the client sends a ‘new game’ request in the middle of a running game:
    1. Server sends ‘general error’ with a ‘try again’ error code, and ends current game
    2. Client can retry ‘new game’ request

MAX GAMES: 10

Notes

- Client plays first
- User-chosen timeout
- Nothing is an ASCII value, example for all values: (1 -> 0b00000001)
- If possible to set the error message, it is strongly recommended to do so

These are not protocol these are error checking on your programs/additional info:

- Stdint.h has the typedefs
  https://pubs.opengroup.org/onlinepubs/009696799/basedefs/stdint.h.html
  - <inttypes.h> has definitions for printf on top of everything stdint.h includes
  - https://stackoverflow.com/questions/7597025/difference-between-stdint-h-and-inttypes-h