

Server Interface Specification for No-Limit

Martin Zinkevich

January 8, 2007

1 Server Responsibilities

Each competitor submits a zipped file. In this file, all of the executable and data files required to run their bot are kept. At the start of a match, this file is transferred to the client machine, unzipped into a directory, and the file `startme.bat` is called with the server IP and port to be connected to, e.g.:

```
startme.bat 128.6.13.13 5555
```

The players connect to the server, and send a version message to indicate the protocol version that they can handle.

The server then maintains the state of the match. At each change of state, both players receive a match state. They can respond by sending a response message, which echoes the state and the action they wish to take.

The game state (which is described in detail later) contains the information about the visible cards and the betting sequence: the player to act, the state of the pots and the bankrolls can be derived from this information. The idea is to make the protocol as simple as possible.

All messages are followed by a carriage return and a line feed < CRLF > (ASCII 13 and ASCII 10).

2 Handshake

The first message sent by the client indicates the version:

$$\langle \text{version} \rangle ::= \text{"VERSION:1.0.0"} \quad (1)$$

3 Match States and Response Messages

A match state consists of a hand number, a seat, and a game state (which is described in the next section). The seat is 1 for the button (who small blinds and bets first on the pre-flop) and 0 for the non-button (who big blinds and bets first on the post-flop, turn, and river).

$$\langle \text{matchstate} \rangle ::= \text{"MATCHSTATE:"} \langle \text{seat} \rangle \text{" "} \langle \text{handNumber} \rangle \text{" "} \langle \text{gamestate} \rangle \quad (2)$$

$$\langle \text{handnumber} \rangle ::= \langle \text{integer} \rangle \quad (3)$$

$$\langle \text{seat} \rangle ::= \text{"0"} | \text{"1"} \quad (4)$$

Match state messages are sent before any player's action, but they are sent to *both* players. In other words, after the flop, when the non-button player is about to check or bet, the button player also receives the state, which may be helpful to the button player if the non-button player takes a long time replying to the flop.

In addition, a match state message is sent after the showdown. There is no mucking, so at this point all cards are revealed.

A response consists of a state to which it is a response and an action. *f* means fold, *c* means check or call, and *r* means bet or raise. If one is betting, one sends *r* followed by the total number of chips one will have in the pot after the bet. This is most easily described through an example. First, let us consider a message sent to Player 0.

`MATCHSTATE:0:392:b1b2r6r18r30c30/r32r96:Ks2d|/5s2s6h`

We will pretend the button player is Bob and the non-button player is Alice. What this message indicates is:

1. **MATCHSTATE:** This message indicates the state of the match. Used such that future protocols may have other information.
2. **0:** This indicates the seat of the player receiving the message. In particular, this player is in seat 0, the non-button player (Alice), the one second to act on the first round, and first to act thereafter.
3. **b1b2r6r18r30c30/** The actions in the first round of betting. Small blinds and big blinds are shown first (b1b2). Then, Bob is first to act. Since he placed the small blind of 1, he first saw Alice's 2 chips of the big blind (adding 1 more chip), and then doubled the size of the pot by raising 4. Thus, he had 6 chips at the end, indicated by r6. Then, Alice saw his 6 chips (adding 4 more chips) and then raised the size of the pot (adding 12 more chips), for 18 total chips at the end, indicated by r18. Then, Bob saw her 18 chips of the non-button player, and made the minimum raise (the size of the previous raise) of 12 chips, bringing his total chips in the pot to 30, indicated by r30. Alice calls, indicated by c30.
4. **r32r96:** This is the second round of betting so far. In particular, Alice bet 2 chips (the minimum size for the first bet on a round, raising her amount in the pot to 32. Then, Bob made a pot raise, calling Alice's 2 chips and then raising 64 chips.
5. **Ks2d|/** These are Alice's hole cards.
6. **5s2s6h** These are the flop cards.

Now, Alice would like to make a pot raise. What she can do is take the 96 sent by Bob and triple it, sending back:

```
MATCHSTATE:0:392:b1b2r6r18r30c30/r32r96:Ks2d|/5s2s6h:r288
```

Then, Bob receives:

```
MATCHSTATE:1:392:b1b2r6r18r30c30/r32r96r288:|QsQd/5s2s6h
```

Note that there is a 1 instead of a 0 (Bob is the button), and |QsQd/ instead of Ks2d|/, the vertical bar being before Bob's cards. Bob calls, sending back:

```
MATCHSTATE:1:392:b1b2r6r18r30c30/r32r96r288:|QsQd/5s2s6h:c
```

The vertical bar makes more sense when we consider the showdown message sent:

```
MATCHSTATE:1:392:b1b2r6r18r30c30/r32r96r288c288/c288c288/c288c288:Ks2d|QsQd/5s2s6h/9c/Ac
```

Thus, Bob wins with two queens. The amount won? The last number in the betting sequence, 288. Suppose that Bob had folded:

```
MATCHSTATE:0:392:b1b2r6r18r30c30/r32r96r288f:Ks2d|/5s2s6h/
```

Then Alice would have won the second-to-last number in the betting sequence, 96. Thus, several key aspects can be easily obtained by the betting sequence.