

Lurk Protocol, version 2.0

The Lurk protocol is intended to support text-based MMORPG-style games, also known as MUDs (Multi-User Dimension). It consists of 13 types of message, some of which are primarily sent by servers and some by clients. Behaviour and game rules are primarily defined by the server, and clients should expect that their character may be updated with different health, location, and wealth at any time. The server is responsible for all computation related to game rules, results of battles, or collecting gold. The client is responsible for communicating with the server and interacting with the player.

The lurk protocol does not specifically contain support for increasing the experience of players. However, the server can allow players to increase their stats. For example, a server could allow an initial stat total of 150, but then for each 1,000 gold collected, allow the player to send an updated set of stats totaling 10 points higher. This behaviour is not required.

Protocol Messages

All protocol message begin with an 8-bit type, followed by 0 or more bytes of further information. The amount of bytes to be read can be determined by the type, and in some cases a message length field further into the message. Messages are listed below by type. Notes:

- Variable-length text fields are sent without a null terminator.
- All numbers are sent little-endian. This makes things easy for x86 users at the expense of being unusual.
- Except as noted (health), all integer fields are unsigned.

MESSAGE

Sent by the client to message other players. Can also be used by the server to send "presentable" information to the client (information that can be displayed to the user with no further processing). Clients should expect to receive this type of message at any time, and servers should expect to relay messages for clients.

Byte	Meaning
0	Type, 1
1-2	Message Length, 16 bits unsigned.
3-34	Recipient Name, 32 bytes total.
35-66	Sender Name, 32 bytes total.

67+ Message. Length was specified earlier.

CHANGEROOM

Sent by the client only, to change rooms. If the server needs to change the room a client is in, it should send an updated room and character message to explain the new location.

Byte	Meaning
0	Type, 2
1-2	Number of the room to change to. The server will send an error if an inappropriate choice is made.

FIGHT

Initiate a fight against monsters. This will start a fight in the current room against the monsters which are presently in the room. Players with the join battle flag set, who are in the same room, will automatically join in the fight. The server will allocate damage and rewards after the battle, and inform clients appropriately. Clients should expect a slew of messages after starting a fight, especially in a crowded room. This message is sent by the client. If a fight should ensue in the room the player is in, the server should notify the client, but not by use of this message.

Byte Meaning

0 Type, 3

PVPFIGHT

Initiate a fight against another player. The server will determine the results of the fight, and allocate damage and rewards appropriately. The server may include players with join battle in the fight, on either side. Monsters may or may not be involved in the fight as well. This message is sent by the client.

Byte Meaning

0 Type, 4

1-32 Name of target player

LOOT

Loot gold from a dead player or monster. The server may automatically gift gold from dead monsters to the players who have killed them, or wait for a LOOT message. The server is responsible for communicating the results of the LOOT to the player, by sending an updated CHARACTER message. This message is sent by the client.

Byte Meaning

0 Type, 5
1-32 Name of target player

START

Start playing the game. A client will send a CHARACTER message to the server to explain character stats, which the server may either accept or deny (by use of an ERROR message). If the stats are accepted, the server will not enter the player into the game world until it has received START. This is sent by the client.

Byte Meaning

0 Type, 6

ERROR

Notify the client of an error. This is used to indicate stat violations, inappropriate room connections, attempts to loot nonexistent or living players, attempts to attack players or monsters in different rooms, etc.

Byte Meaning

0 Type, 7
1 Error code. List is given below.
2-3 Error message length.
4+ Actual error message, of the specified length.

Error codes:

Code	Meaning
0	Other (not covered by any below error code)
1	Bad room. Attempt to change to an inappropriate room
2	Player Exists. Attempt to create a player that already exists.
3	Bad Monster. Attempt to loot a nonexistent or not present monster.
4	Stat error. Caused by setting inappropriate player stats.
5	Not Ready. Caused by attempting an action too early, for example changing rooms before sending START or CHARACTER.
6	No target. Sent in response to attempts to loot nonexistent players, fight players in different rooms, etc.
7	No fight. Sent if the requested fight cannot happen for other reasons (i.e. no live monsters in room)
8	No player vs. player combat on the server. Servers do not have to support player-vs-player combat.

ACCEPT

Sent by the server to acknowledge a non-error-causing action which has no other direct result. This is not needed for actions which cause other results, such as changing rooms or beginning a fight. It should be sent in response to clients sending messages, setting character stats, etc.

Byte	Meaning
0	Type, 8
1	Type of action accepted.

ROOM

Sent by the server to describe the room that the player is in. This should be an expected response to CHANGEROOM or START. Can be re-sent at any time, for example if the player is teleported or falls through a floor. Outgoing connections will be specified with a series of CONNECTION messages. Monsters and players in the room should be listed using a series of CHARACTER messages.

Byte	Meaning
0	Type, 9
1-2	Room number. This is the same room number used for CHANGEROOM
3-34	Room name, 32 bytes in length
35-36	Room description length
37+	Room description. This can be shown to the player.

CHARACTER

Sent by both the client and the server. The server will send this message to show the client changes to their player's status, such as in health or gold. The server will also use this message to show other players or monsters in the room the player is in or elsewhere. The client should expect to receive character messages at any time, which may be updates to the player or others.

The client will use this message to set the name, description, attack, defense, regen, and flags when the character is created. It can also be used to reprise an abandoned or deceased character.

Byte	Meaning
0	Type, 10
1-32	Name of the player
33	Flags: Starting from the highest bit, Alive, Join Battle, Monster, Started, Ready. The lowest three are reserved for future use.
34-35	Attack

36- Defense
 37
 38- Regen
 39
 40- Health (signed)
 41
 42- Gold
 43
 44- Current room number (may be unknown to the player)
 45
 46- Description length
 47
 48+ Player description

Meaning of flags:

Flag	Meaning
Alive	Character is alive (1 = alive, 0 = not alive)
Join Battle	Character will automatically join battles in the room they are in (1 = join battles, 0 = do not join battles)
Monster	Character is a monster (1 = monster, 0 = player)
Started	Character has started the game (1 = started, 0 = not started)
Ready	Character is ready to start the game (1 = started, 0 = not started)

When a client uses CHARACTER to describe a new player, the server may (should) ignore the client's initial specification for health, gold, and room. The monster flag is used when describing monsters found in the game rather than other human players.

GAME

Used by the server to describe the game. The initial points is a combination of health, defense, and regen, and cannot be exceeded by the client when defining a new character. The stat limit is a hard limit for the combination for any player on the server regardless of experience. If unused, it should be set to 65535, the limit of the unsigned 16-bit integer. This message will be sent upon connecting to the server, and not re-sent.

Byte	Meaning
0	Type, 11
1-2	Initial Points
3-4	Stat limit
5-6	Description Length
7+	Game description

LEAVE

Used by the client to leave the game. This is a graceful way to disconnect. The server never terminates, so it doesn't send LEAVE.

Byte Meaning

0 Type, 12

CONNECTION

Used by the server to describe rooms connected to the room the player is in. The client should expect a series of these when changing rooms, but they may be sent at any time. For example, after a fight, a secret staircase may extend out of the ceiling enabling another connection.

Byte

Meaning

0 Type, 13

1-2 Room number. This is the same room number used for CHANGEROOM

3-34 Room name, 32 bytes in length

35-36 Room description length

37+ Room description. This can be shown to the player.

Fight Calculations

Consider the following player and monster:

Name: Trudy

Description: Black Hat Security Expert

Gold: 10

Attack: 60

Defense: 30

Regen: 10

Status: ALIVE

Location: Broom Closet

Health: 100

Started: YES

Name: Glog

Description: A slimy and toothy character

Gold: 50

Attack: 20

Defense: 30

Regen: 100

Monster

Health: 100

Trudy may or may not be able to defeat this monster, depending on the way the server calculates fight results. Suppose for example that in fighting each sustains the attack of the other in damage, minus the defense. In this case, Trudy receives no damage at all (20 attack from Glog, minus Trudy's 30 defense, leaves no damage). Glog receives 30 damage (60 attack from Trudy, minus 30 defense from Glog). However, Glog has a very high regen of 100. If the server allocates 10\% of the regen to health per turn,

than Glog will receive a 10 point health increase, and Trudy will receive only 1 point increase. Still, in a few turns, Trudy defeats Glog. If, on the other hand, the server were to allocate 30\% of regen as gained health each turn, neither would defeat the other.

The exact method of calculation is a game design decision made at the server level. Clients should not attempt to calculate damage themselves, but expect to receive a CHARACTER message after each fight their player is involved in.

Notes

- Many aspects of the game are not controlled by the protocol, but by the way the server implements it. For example, the result of a fight is determined by the server. The regen stat is available for the server to use, but its use is not required. Connections between rooms do not have to be bidirectional. Monsters do not have to remain in a single room, but can move around and enter the room the player is in (CHARACTER should be sent in this case). Monsters can attack the player, or ignore the player until attacked themselves. Each server can manage a very different game.
- Clients must listen for input both from the user and from the server.
- The user should not be expected to remember everything the server has presented.