

# Ripper-Protocol

*by Johan Hedenby and Mateo Olofsson Falla*

The server-client connection is using TCP/IP. The messages between the applications (server and client) are based on a byte by byte protocol. This is the protocol to interpret those bytes.

Every message starts with an 8 byte long header, followed by data of varied length, or in some cases, no data at all (only header).

Wherever multiple bytes are used to represent a single number, Big-Endian is used (the most significant byte is on the lowest index value).

## Header

In the picture below we have a representation of the header. Which are divided into different squares and each square represents one byte.

Index	Color	Name	Description
0	Red	Start byte	Always the byte 0xFF.
1-4	Blue	Sequence number	Enumerates the messages so the Server and Client knows the order of the messages.
5	Green	Message type	Indicates which data-protocol should be used. See the Data section for each message type.
6-7	Purple	Data length	The number of bytes in the data.



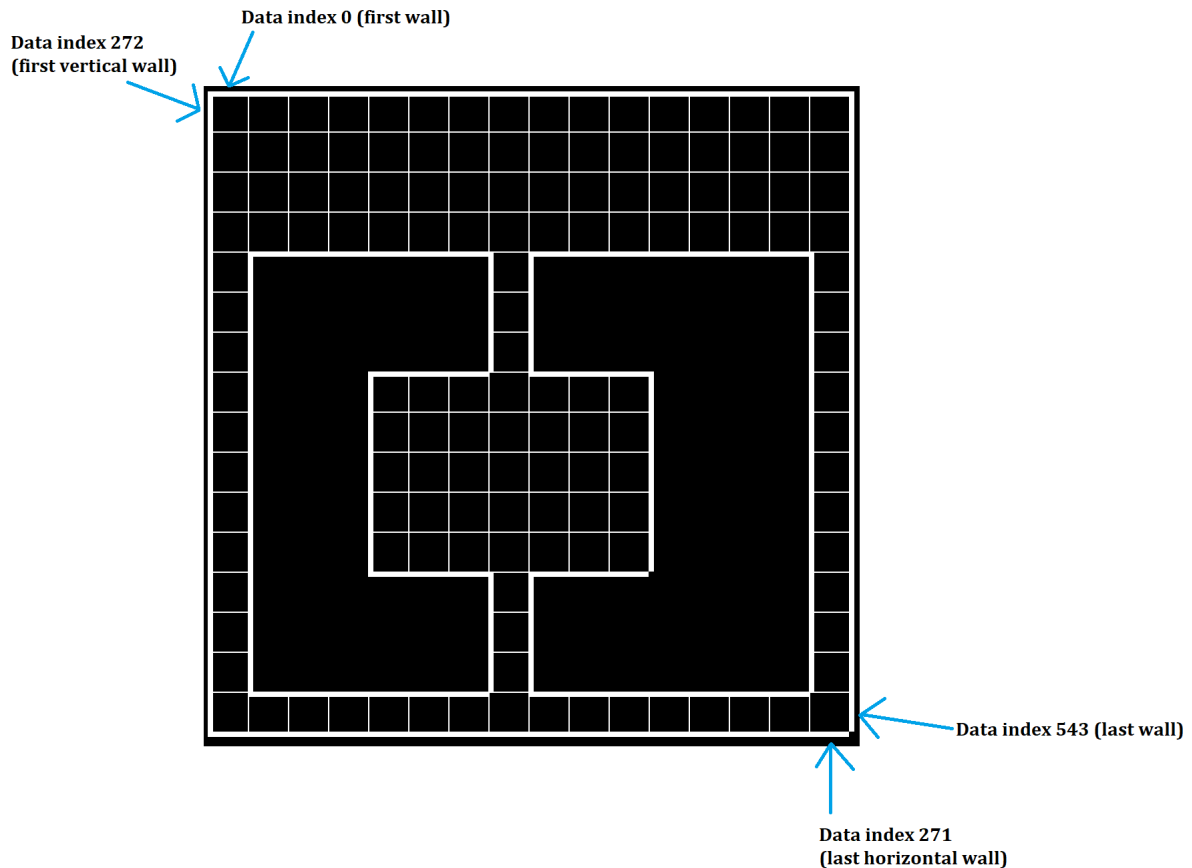
# Data

Different types of messages have different protocols to interpret the data. The type of the message is determined by the message type byte in the header. Messages from the server to the clients are numbered below 100. Messages from the client to the server are numbered from 100 and above. As mentioned before, some message types do not need any data (only header). For example when the client asks for a refresh, as the message type in the header is enough to indicate this. Then the data length will be 0.

Message Type	Name	Description
1	Walls	See Walls
2	Floor	See Floors
3	Inventory	See Inventory
4	Combat Information	See Combat Information
5	Request Skill Choice	Server asks the client which skill the player wants to level up. Data length 0 (only header). The client answers with a "Answer Skill Choice".
6	Waiting For Other Players	The server indicates to one client that the other client is not ready. Data length 0 (only header).
7	Game Over	See Game Over
100	Direction Key	See Direction key
101	Skill Key	See Skill key
102	Use Item	See Use Item
103	Refresh	This indicates to the server that the client wants to refresh all current information. Data length 0 (only header).
104	Answer Skill Choice	See Answer Skill Choice
105	Start Game	Indicates to the Server that the Client is ready. The Data contains the player's name. Data length can vary, depending on the length of the name.
106	Quit	Indicates to the Server that the Client has stopped the Game. Data length 0 (only header).

**Walls:**

- Each byte in the data represents a wall on the game board. The game board has 16x17 horizontal walls and 17x16 vertical walls. The data starts with horizontal walls, followed by vertical walls. Always reading left to right and secondly from up to down. For example the first byte in the data represents the horizontal wall on the upper left corner, the next byte represents the horizontal wall on the right side of the first wall, and so on.



- In this Message type the Server can send different types of values to indicate different types of walls, as shown below:
  - 1 = No wall, outside playable space
  - 2 = No wall, inside playable space
  - 3 = Wall
  - 4 = Door

**Floor:**

- Each byte in the data represents a floor tile on the game board. The game board has 16x16 tiles. Always reading left to right and secondly from up to down. For example the first byte in the data represents the tile on the upper left corner, the next byte represents the tile on the right side of the first tile, and so on. Depending on the byte value of each tile, the client displays an image corresponding to that value on the tile. The image could be of players, monsters, items or no image at all.

Table of floor tile types:

Byte Value:	Name:	Description:
1	Empty	No floor, outside playable space
2	Empty	Floor (empty)
4	Player 1	An image of Player 1
5	Player 2	An image of Player 2
6	Monster 1	An image of Monster 1
7	Monster 2	An image of Monster 2
8	Monster 3	An image of Monster 3
9	Monster 4	An image of Monster 4
10	Monster 5	An image of Monster 5
11	Monster 6	An image of Monster 6
12	Monster-Boss 1	An image of Monster-Boss 1
13	Monster-Boss 2	An image of Monster-Boss 2
14	Monster-Boss 3	An image of Monster-Boss 3
15	Potion	An image of Potion
16	One Handed-Sword	An image of One Handed-Sword
17	Two Handed-Sword	An image of Two Handed-Sword
18	Light-Armor	An image of Light-Armor
19	Medium-Armor	An image of Medium-Armor
20	Heavy-Armor	An image of Heavy-Armor
21	Shield	An image of Shield

### Inventory:

- The data contains 12 strings, each 35 characters long (35 bytes). The data length is always 420 bytes long (12\*35). Each string represents an item in the inventory.

### Combat Information:

- The combat information from the server to the client, contains info about the players HP, the players skill induction progress, the HP of the ally and depending on whether the player is fighting a monster or not, the data may also contain info about the monster. The data length is either 6 or 45 bytes long. The picture below displays the bytes.



Name	Color:	Byte Length:	Description:
Player's Health	Red	2	The current HP of the player.
Induction bar	Purple	1	The current induction time of used skill in percent.
Other Players's Health	Green	2	The current HP of the other player.
Show monster Info (On / Off)	Grey	1	If this byte is value 0, the player is not fighting any monster, no monster info is included in this data.  If the byte value is 1, the player is fighting. The next 39 bytes include info about the targeted monster.
Monster's Health	Light-Blue	2	The current HP of the monster that the player is fighting with.
Monster's Max Health	Dark-Blue	2	The Max HP the monster has.
Monster's Name	Brown	35	The name of the monster.

### Game Over:

- The game over data will contain 11 strings, where the first string is 35 bytes long and reserved for the title, for example "Game over" or "Victory". The remaining 10 strings are 100 bytes each and represent a line under the title. The data length is 1035 bytes.

**Direction key:**

- A message from the client to the server, indicating what direction the player wants to go. The data length is always 1 byte long.

Name:	Value:	Direction:
Right	1	Move player to the Right
Left	2	Move player to the Left
Up	3	Move player Up
Down	4	Move player Down

**Skill key:**

- A message from the client to the server, indicating what skill the player wants to use. The data length is always 1 byte long.

Name:	Value:	Description
Potion	1	Use potion.
Taunt	2	Use Taunt.
Protector	3	Use Protector.
Sword-Master	4	Use Sword-Master.
Play-Nice	5	Use Play-Nice.
Aggressive Attack	6	Use the stance of Aggressive Attack
Normal Attack	7	Use the stance of Normal Attack
Defensive Attack	8	Use the stance of Defensive Attack

**Use Item:**

- A message from the client to the server, indicating what item the player wants to use or drop. The first byte represents use (value 1) or drop (value 2). The second byte represents the item position in the inventory, starting from position 1 (not index 0). Position 1 will always be the player's potions.

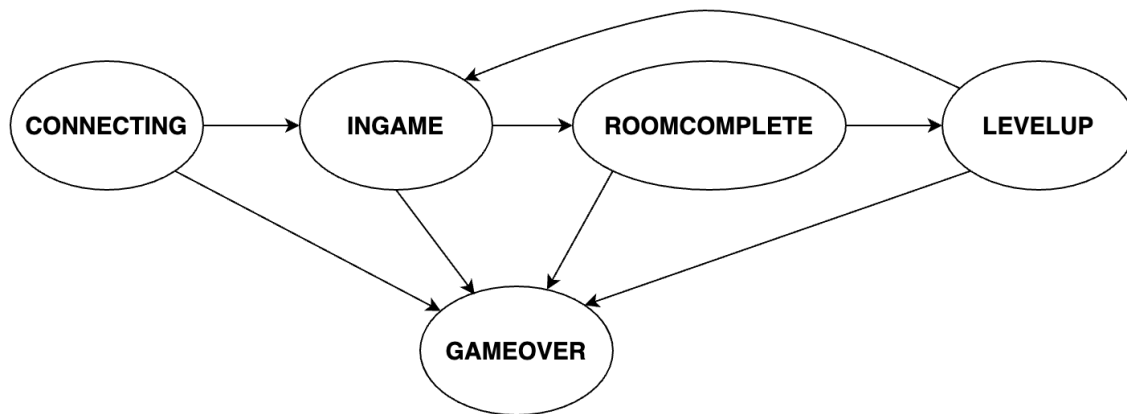
**Answer Skill Choice:**

- A message from the client to the server, as an answer to a "Request Skill Choice"-message. The player chooses a skill to level up. The data length is always 1 byte long.

<b>Name:</b>	<b>Byte Value:</b>	<b>Description:</b>
Taunt	1	Level up Taunt
Protector	2	Level up Protector
Sword-Master	3	Level up Sword-Master
Play-Nice	4	Level up Play-Nice

# State-Diagram

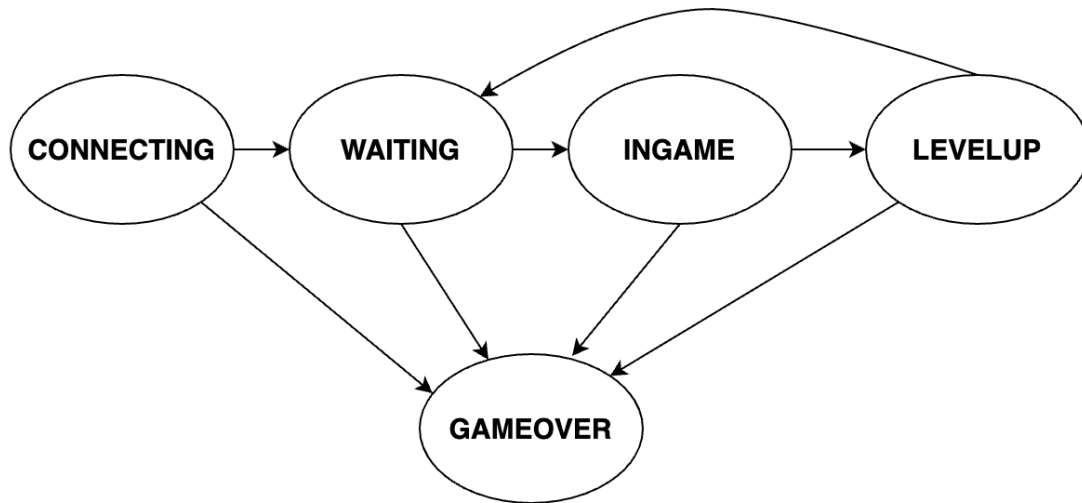
## Server - Game state:



State	Next States	Description
CONNECTING	INGAME, GAMEOVER	When both Players are in WAITING stage CONNECTING → INGAME If Server or Client end game CONNECTING → GAMEOVER
INGAME	ROOMCOMPLETE, GAMEOVER	When all monsters are dead on the current room INGAME → ROOMCOMPLETE If Server or Client end game INGAME → GAMEOVER
ROOMCOMPLETE	LEVELUP, GAMEOVER	When both Players are in LEVELUP stage ROOMCOMPLETE → LEVELUP If Server or Client end game ROOMCOMPLETE → GAMEOVER
LEVELUP	INGAME, GAMEOVER	When both Players are in WAITING stage LEVELUP → INGAME If Server or Client end game LEVELUP → GAMEOVER
GAMEOVER		Game ends.

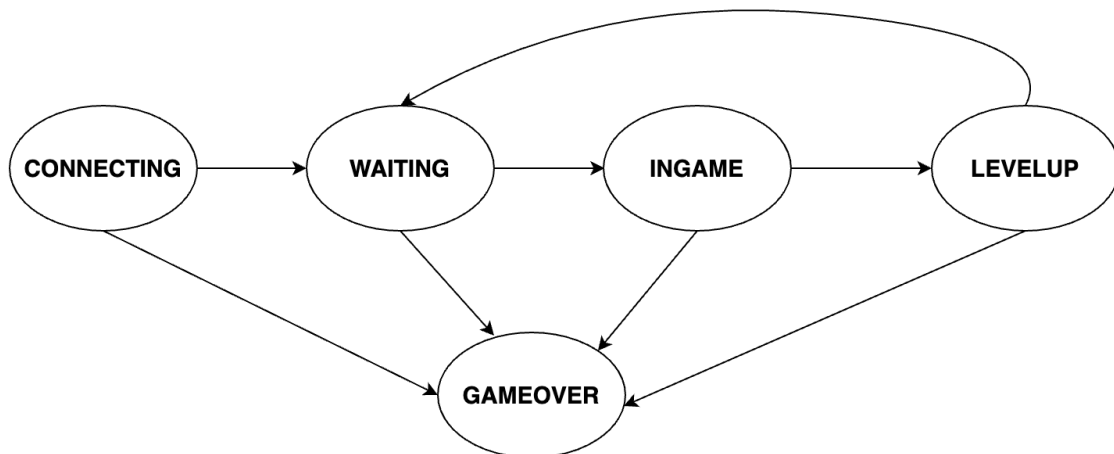


## Server - Each players state:



State	Next State	Description
CONNECTING	WAITING, GAMEOVER	When Client sends Start Game to Server CONNCTING → WAITING If Server or Client end game CONNCTING → GAMEOVER
WAITING	INGAME, GAMEOVER	When Server sends Wall WAITING → INGAME If Server or Client end game WAITING → GAMEOVER
INGAME	LEVELUP, GAMEOVER	When Server sends Request Skill Choice INGAME → LEVELUP If Server or Client end game INGAME → GAMEOVER
LEVELUP	WAITING, GAMEOVER	When Server receives Answer Skill Choice from Client LEVELUP → WAITING If Server or Client end game LEVELUP → GAMEOVER
GAMEOVER		Game ends.

## Client-State:



State:	Next State:	Description:
CONNECTING	WAITING, GAMEOVER	When Client sends Start Game to Server CONNECTING → WAITING. If Server or Client end game CONNECTING → GAMEOVER.
WAITING	INGAME, GAMEOVER	When Client receives Walls from Server WAITING → INGAME. If Server or Client end game WAITING → GAMEOVER.
INGAME	LEVELUP, GAMEOVER	When Client receives Request Skill Choice from Server INGAME → LEVELUP. If Server or Client end game WAITING → GAMEOVER.
LEVELUP	WAITING, GAMEOVER	When Client sends Answer Skill Choice to Server LEVELUP → WAITING If Server or Client end game LEVELUP → GAMEOVER.
GAMEOVER	NO NEW STATE	Game ends.