

R-Type Protocol

Status of This Memo

This memo defines an Experimental Protocol for the internet community. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

R-Type is a single player shoot-them-up type of game, only playable offline.

This document describes how R-Type Protocol (RTP) must be implemented to add online multiplayer option to the game.

The RTP protocol is a binary-based protocol, with the simplest client being any socket program capable of connecting to the server.

Table of Contents

1. INTRODUCTION	4
1.1 Server	4
1.2 Clients	4
1.3 Rooms	4
2. THE RTP SPECIFICATION	4
2.1 Overview	4
2.2 Packet	4
2.2.1 Header	4
2.2.2 Data	5
3. RTP CONCEPT	5
3.1 Pre-game	5
3.1.1 Connection	5
3.1.2 Disconnection	5
3.1.3 Room	6
3.1.3.1 Join	6
3.1.3.2 Leave	6
3.1.3.3 Rooms	6
3.1.3.4 Players	6
3.1.3.5 Ready	6
3.1.3.6 Player info	6
3.1.3.7 Files	6
3.1.3.9 Start	7
3.1.4 Example	7
3.2 In-game	8
3.2.1 Prelude	8
3.2.2 Event	8
3.2.2.1 Movement	8
3.2.2.2 Leave	8
3.2.3 Object	9
3.2.3.1 Players	9
3.2.3.2 Enemies	9

3.2.3.3	Bullets	9
3.2.3.4	Bonus	9
3.2.4	Collisions	9
3.2.5	Scoring	9
3.2.6	Game Over	9
3.2.7	Postlude	10
4.	PACKET DETAILS	10
4.1	Header possibilities	10
4.1.1	Size	10
4.1.2	Type	10
4.1.3	Transaction Type	10
4.1.4	Sub-type	10
4.1.4.1	Connection	10
4.1.4.2	Room	11
4.1.4.3	Game	11
4.2	Connection	11
4.2.1	Client	11
4.2.2	Server	11
4.2.2.1	Success	12
4.2.2.2	Error	12
4.3	Room operations	12
4.3.1	Disconnection	12
4.3.1.1	Client	12
4.3.2	Join	12
4.3.2.1	Client	13
4.3.2.2	Server	13
4.3.2.2.1	Success	13
4.3.2.2.2	Error	13
4.3.3	Leave	13
4.3.3.1	Client	14
4.3.3.2	Server	14
4.3.3.2.1	Success	14
4.3.3.2.2	Error	14
4.3.4	Rooms	14
4.3.4.1	Client	15
4.3.4.2	Server	15
4.3.5	Players	15
4.3.5.1	Client	15
4.3.5.2	Server	15
4.3.5.2.1	Success	16
4.3.5.2.2	Error	16
4.3.6	Ready	16
4.3.6.1	Client	16
4.3.6.2	Server	16
4.3.6.2.1	Success	17
4.3.6.2.2	Error	17
4.3.7	Player Info	17
4.3.7.1	Client	17
4.3.7.2	Server	17
4.3.8	File	17
4.3.8.1	Server	18
4.3.8.2	Client	19
4.3.9	Start	19

4.3.9.1	Client	19
4.3.9.2	Server	20
4.4	Prelude	20
4.5	Game operations	20
4.5.1	Events	20
4.5.1.1	Movement	20
4.5.1.1.1	Client	20
4.5.1.1.1	Server	21
4.5.1.2	Leave	21
4.5.1.2.1	Client	21
4.5.1.2.1	Server	21
4.5.2	Objects	21
4.5.2.1	Creation	21
4.5.2.2	Modification	22
4.5.2.3	Destruction	22
4.5.3	Player's bullets	22
4.5.3.1	Client	22
4.5.3.1	Server	23
4.5.4	Scoring	23
4.5.5.1	Server	23
4.5.5	Game Over	23
4.5.6.1	Server	23
4.6	Postlude	24
4.6.1	Server	24
5	TROUBLESHOOT	24
5.1	Latency	24
6	CONTACTS	24

1. INTRODUCTION

The RTP (R-Type Protocol) has been designed over a number of years to add an online multiplayer dimension to the game based on a client-server architecture. This document describes the current RTP.

The RTP has been developed on systems using the TCP/IP and UDP network protocol. In which TCP/IP will be used for all the settings of the game and important packets, as for the UDP to exchange informations in-game.

R-Type itself is a side scrolling shoot them up arcade game produced 1987. The player controls a space fighter to defend humanity against a mysterious but powerful alien life-form known as "Bydo".

1.1 Server

The server host up to ten rooms where up to four players can join. See the room section for more information.

1.2 Clients

Each client is distinguished from other clients by a unique nickname having a maximum length of (11) characters. See the protocol grammar rules for what may or may not be used in a nickname.

1.3 Rooms

The rooms are kind of a lobby in which players can join a game. It MUST allow a maximum of four players. To start a game, everyone in the room MUST set there status to ready (More precision in the chapter 3.1 pre-game).

2. THE RTP SPECIFICATION

2.1 Overview

The protocol as described herein is for client to server connections.

2.2 Packet

2.2.1 Header

The header MUST be the first two bytes of each packet send to the server. To be valid the RTP packet header is REQUIRED to be as described below and each number representing a bit:

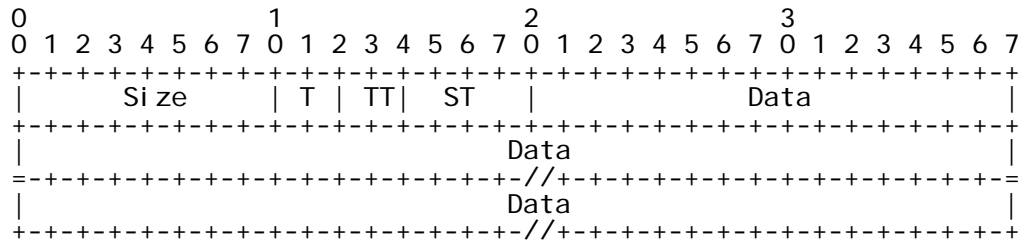


Figure 1: Format Size of a Packet

Size: it is the packet size including the data (8 bits).
T: the type define what kind of packet it is (2 bits).
TT: with this you can set the type of transaction of the packet (2 bits).
ST: it is the sub type of the packet (4 bits).
Data: here is all the data you want to send to the server (254 * 8 bits).

2.2.2 Data

The data contains all the informations the sender want to transmit to the receiver. It MUST have at the maximum 254 bytes.

3. RTP CONCEPT

The RTP is composed by two parts: a pre-game one and an in-game one. We are going to introduce these two parts to see the concept used in it. We will not introduce the packet contents but only the concept of the several steps.

3.1 Pre-game

The pre-game can be splitted in three parts. All the communication will be done using TCP/IP protocol.

3.1.1 Connection

First, the player MUST connect his client to the server. He is REQUIRED to use the server's IP and port. He will be able to join the server only if he specifies a valid login (11 characters).

```
<login> ::= [ a-z | A-Z | _ ][ a-z | A-Z | _ | 0-9]*
```

If the login choosen by the player exists, the server will asks him to choose another login. If it doesn't respect the BNF of the login the server will notify the client about the wrong login.

3.1.2 Disconnection

The client disconnect himself from the server by notifying it.

3.1.3 Room

The rooms are the heart of the game. These are the different actions that can do a client.

3.1.3.1 Join

A client MAY join a room only if there is enough place. To remind you, a room MUST NOT accept more than four clients. He can try to go into a room, the server will reply if he can join this room.

3.1.3.2 Leave

A client MAY leave a room only if not everybody is in state ready (for more information about the state, see the next part).

3.1.3.3 Rooms

A client can ask to the server to list the rooms available. He will receive the list of the rooms.

3.1.3.4 Players

A client can get the list of the players who are in a room.

3.1.3.5 Ready

Every client in a room is linked to a state. You have two states: ready and not ready. If everyone in the room is ready, then the game get started. When a game is about to begin, all clients cannot change their states.

3.1.3.6 Player info

Every objects, the players and the bullets are unique. They are all associated with a unique ID by the server to distinguished each of them. The port, on which they will bind their UDP socket, will be sent with the ID of each client.

For each game, ids from 0 to 3 are reserved to the four players in game. It is the first information sent by the server to the client when every players in the room is ready.

3.1.3.7 Files

Files are required to play the game in the client side. To be sure that it has them the server will ask him and if not send them over to him.

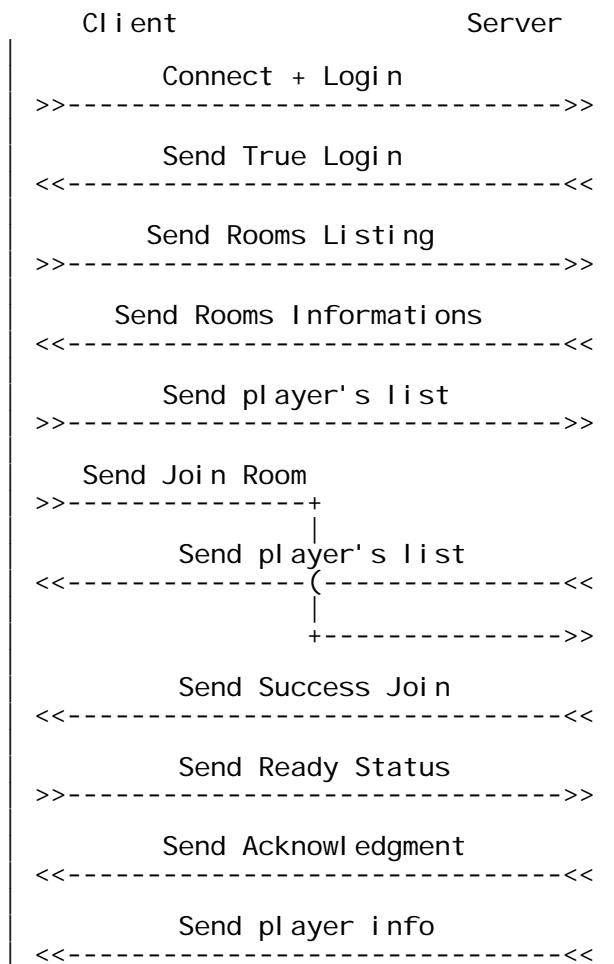
3.1.3.10 Start

If the client has received all the information about the game, he MUST notify the server that he is ready to start the game. If he does not notify the server, the game will never start.

In fact, if the server does not receive all the responses from the different players, it will never launch the game. The clients will be informed of the beginning of the game when all of them are ready to start.

3.1.4 Example

This scheme is an example that show a classic use of a player before a game.



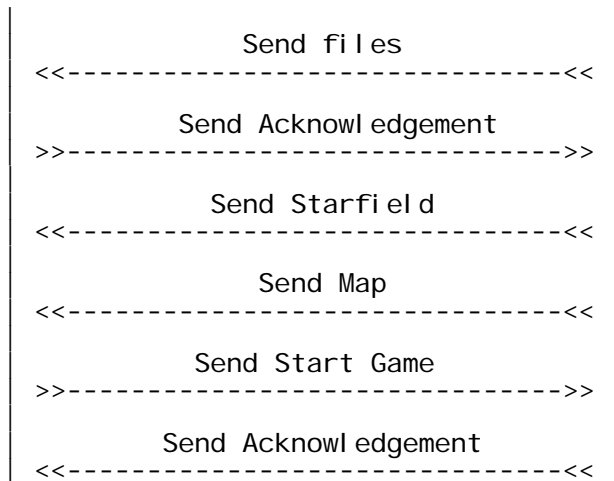


Figure 1: Initialization of the connection and prelude of a game.

3.2 In-game

This section is devoted to describe the actual organization made between the client and the server during the game.

3.2.1 Prelude

It's in this phase that the clients will receive the informations about the ships they will play with.

3.2.2 Event

When the clients have received all the informations, they will be able to act in the level in which they are. Let us see what they are allowed to do.

3.2.2.1 Movement

A player MAY move in the level. He MUST send to the server the direction he took. There is two messages to send. One when he begin to push the button and one when he released this one. He MAY go up, down, left or right. That move is calculated by the server and sent back to all the players in the game.

3.2.2.2 Leave

A player MAY leave a game when he wants to. He would be a coward in the eyes of his/her friend who played with him of course. He will be redirected out of the room he is in and invited to chose another one room.

3.2.3 Object

All the following things described below are all considered as object.

3.2.3.1 Players

A player is a object which is controle by one of the player. It will change of behavior by the several events emits by the different player.

3.2.3.2 Ennemi es

R-Type is not a pushover. Indeed, you will find in your way a horde of ennemies who only want one thing: kill you and your friends. You have two options to survive. Kill all the monsters you will cross in your way or avoid their attacks by passing through the crowd.

Be carefull, a simple contact with an ennemy's bullet or this one will kill you directly (see the part below this one on collision).

3.2.3.3 Bul lets

They should use their weapon to shoot their ennemies. A player may enhance the power of the bullets by keeping the shoot's key pushed. The more you do, the more the players weapon's power will increase. Each bullet is unique and linked to a particular player. The ennemy can shoot the player too with their own bullets.

3.2.3.4 Bonus

A player can get bonuses. They can appear by killing ennemies or just randomly in the middle of the game.

3.2.4 Collision

A player is very vulnerable. A contact with an ennemy or with some elements of the level (for example a space rock...) will kill him. A client will be notified by the server when he will be touched by one of those elements. You MUST be carefull!

3.2.5 Scoring

R-Type is a shoot them up, a system of scoring is of course necessary to add more competitiveness between players and much more fun.

To get your score increased, you have to kill monsters or get the bonuses on the map.

3.2.6 Game Over

Like it has been explain previously, a simple contact with the map, an ennemy or his bullets consume one life. When all the players have no life left, the game is over.

3.2.7 Postlude

When the boss dies that means that the players have won and so they can load the next level.

4. PACKET DETAILS

On the following pages are descriptions of each packet recognized by the server and client. All packets described in this section must be implemented by any server for this protocol.

The server to which a client is connected is required to handle the complete packet, returning any appropriate errors. If the server encounters a fatal error while handling a message, an error must be sent back. A fatal error may be considered to be an incorrect packet.

4.1 Header possibilities

Herein you will find a description of all settings you can use for the packet's header.

4.1.2 Type

Type MUST be one of the following:

RT_TYPE_CONNECTION (0)
RT_TYPE_ROOM (1)
RT_TYPE_GAME (2).

4.1.3 Transaction Type

The transaction type (TT) MUST be one of the following:

RT_TT_REQUEST (0)	RT_TT_CREATION (0)
RT_TT_REPLY (1)	RT_TT_MODIFICATION (1)
RT_TT_ACK (2)	RT_TT_DESTRUCTION (2)
RT_TT_ERROR (3)	

4.1.4 Sub-type

Each Type have their own sub-types.

4.1.4.1 Connection

CONNECTION type is REQUIRED to have at least those sub-types:

RT_SUB_AUTH (0)
RT_SUB_DISC (1)
RT_SUB_ALIVE (2)

4.1.3.2 Room

ROOM is REQUIRED to have at least those sub-types:

```
RT_SUB_JOIN (0)
RT_SUB_LEAVE (1)
RT_SUB_ROOMS (2)
RT_SUB_PLAYERS (3)
RT_SUB_READY (4)
RT_SUB_ID (5)
RT_SUB_STARFIELD (6)
RT_SUB_MAP (7)
RT_SUB_START (8)
RT_SUB_FILE (9)
```

4.1.4.3 Game

GAME is REQUIRED to have at least the following sub-types:

```
RT_SUB_EVENT (0)
RT_SUB_OBJECT (1)
RT_SUB_BULLET (2)
RT_SUB_SCORING (3)
RT_SUB_GAMEOVER (4)
RT_SUB_LEVELOVER (5)
RT_SUB_STARFIELD (6)
```

4.2 Connection

A player who wants to join a server MUST introduce himself by sending in a packet with his nickname (it must respect the BNF).

4.2.1 Client

Size: Depending on the size of the data
Type: RT_TYPE_CONNECTION
Transaction Type: RT_TT_REQUEST
Sub-Type: RT_SUB_AUTH
Data: it will contain the login

Size	T	TT	ST	Data
12(bytes)max	0	0	0	////////////////////////////////////

4.2.3 Server

Depending on the situation the server can reply one of the following packets.

4.2.3.1 Success

Size: 0
 Type: RT_TYPE_CONNECTION
 Transaction Type: RT_TT_ACK
 Sub-Type: RT_SUB_AUTH
 Data: none

Size	T	TT	ST	Data
0	0	2	0	

4.2.3.1 Error

Size: Depending on the size of the data
 Type: RT_TYPE_CONNECTION
 Transaction Type: RT_TT_ERROR
 Sub-Type: RT_SUB_AUTH
 Data: it will contain one of the following errors:
 'The login you sent is not in the good format.'

Size	T	TT	ST	Data
254(bytes)max	0	3	0	////////////////////////////////////

4.3 Room operations

A player is able to do some actions within the rooms. Here are the details of them.

4.3.1 Disconnection

A player who wants to leave the server MUST notify him by sending in a packet with his nickname.

4.3.1.1 Client

Size: 0
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REQUEST
 Sub-Type : RT_SUB_DISCONNECT
 Data: none

Size	T	TT	ST	Data
0	0	0	9	

4.3.2 Join

This is what a player MUST send to join a room.

4.3.2.1 Client

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REQUEST
 Sub-Type: RT_SUB_JOIN
 Data: it will contain the name (253 bytes max.) of the room

Size	T	TT	ST	Data
254(bytes)max	1	0	0	////////////////////////////////////

4.3.2.2 Server

Depending on the situation the server can reply one of the following packets.

4.3.2.2.1 Success

Size: 0
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_ACK
 Sub-Type: RT_SUB_JOIN
 Data: none

Size	T	TT	ST	Data
0	1	2	0	

4.3.2.2.2 Error

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_ERROR
 Sub-Type: RT_SUB_JOIN
 Data: it will contain one of the following errors:
 'The room is full.'
 'Not an existing room.'
 'The room is currently in game.'

Size	T	TT	ST	Data
254(bytes)max	1	3	0	////////////////////////////////////

4.3.3 Leave

The player MUST send this packet to leave the room he is in.

4.3.3.1 Client

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REQUEST
 Sub-Type: RT_SUB_LEAVE
 Data: it will contain the name (253 bytes max) of the room

Size	T	TT	ST	Data
254(bytes)max	1	0	1	////////////////////////////////////

4.3.3.1 Server

Depending on the situation the server can reply one of the following packets.

4.3.3.1.1 Success

Size: 0
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_ACK
 Sub-Type: RT_SUB_LEAVE
 Data: none

Size	T	TT	ST	Data
0	1	2	1	

4.3.3.1.2 Error

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_ERROR
 Sub-Type: RT_SUB_LEAVE
 Data: it will contain one of the following errors:
 'Not an existing room.'
 'You're not currently in a room.'

Size	T	TT	ST	Data
254(bytes)max	1	3	1	////////////////////////////////////

4.3.4 Rooms

When a player wants the list of available rooms he MUST send this.

4.3.4.1 Client

Size: 0
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REQUEST
 Sub-Type: RT_SUB_ROOMS
 Data: none

Size	T	TT	ST	Data
0	1	0	2	

4.3.4.2 Server

Depending on the situation the server can reply one of the following packets.

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REPLY
 Sub-Type: RT_SUB_ROOMS
 Data: it will contain the list of the rooms name. The format is a string in which each room's name will be separated by the character '\0'.

Size	T	TT	ST	Data
254(bytes)max	1	1	2	////////////////////////////////////

4.3.5 Players

When a player wants the list of available players in a room he MUST send this.

4.3.5.1 Client

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REQUEST
 Sub-Type: RT_SUB_PLAYERS
 Data: it will contain the room's name.

Size	T	TT	ST	Data
254(bytes)max	1	0	3	////////////////////////////////////

4.3.5.2 Server

Depending on the situation the server can reply one of the following.

4.3.5.2.1 Success

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_REPLY

Sub-Type: RT_SUB_PLAYERS

Data: it will contain the list of the players login. The format is a string in which each player's login(11 bytes max) will be separated by the character '\0'.

Size	T	TT	ST	Data
48(bytes)max	1	1	3	////////////////////////////////////

4.3.5.2.1 Error

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_ERROR

Sub-Type: RT_SUB_PLAYER

Data: this data will contain one of the following errors:
'This room doesn't exists.'

Size	T	TT	ST	Data
254(bytes)max	1	3	3	////////////////////////////////////

4.3.6 Ready

When a player is ready to play he MUST send the following.

4.3.6.1 Client

Size: 0

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_REQUEST

Sub-Type: RT_SUB_READY

Data: none

Size	T	TT	ST	Data
0	1	0	4	

4.3.6.2 Server

Depending on the situation the server can reply one of the following packets.

4.3.6.2.1 Success

Size: depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REPLY
 Sub-Type: RT_SUB_READY
 Data: the login of the player who is ready

Size	T	TT	ST	Data
11(bytes)max	1	1	4	

4.3.6.2.2 Error

Size: depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REPLY
 Sub-Type: RT_SUB_READY
 Data: this data will contain one of the following errors:
 'You are not in a room.'
 'Too late to be un-ready.'

Size	T	TT	ST	Data
254(bytes)max	1	3	4	////////////////////

4.3.7 Player info

When all the players are known as ready the server MUST send them their id and the port. Here is how it MUST be done.

4.3.7.1 Server

Size: Depending on the size of the data
 Type: RT_TYPE_ROOM
 Transaction Type: RT_TT_REPLY
 Sub-Type: RT_SUB_ID
 Data: it will contain the port (2 bytes) and the id (4 bytes) of the player and his login.

Size	T	TT	ST	Data
254(bytes)max	1	1	5	////////

4.3.8 Files

Before sending the map and the starfield, the server needs to send the file REQUIRED to play the game. Here is how it will do it.

4.3.8.1 Server

The server MUST ask to the client if he has the file needed by sending this.

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_CREATE

Sub-Type: RT_SUB_FILE

Data: it will contain the file size (4 bytes) and the file name.

Size	T	TT	ST	Data
254(bytes)max	1	0	9	////////////////////////////////////

If he has a positive answer, he will continue by the sending the content with this:

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_MODIFICATION

Sub-Type: RT_SUB_FILE

Data: it will contain the content of the file.

Size	T	TT	ST	Data
254(bytes)max	1	1	9	////////////////////////////////////

The below packet announced the end of transfer.

Size: 0

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_MODIFICATION

Sub-Type: RT_SUB_FILE

Data: none

Size	T	TT	ST	Data
0	1	1	9	

If the client has already the file, the server will send the next one. After the server has sent all the needed files it MUST send this to notify the end of the file transfer phase.

Size: 0

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_DESTRUCTION

Sub-Type: RT_SUB_FILE

Data: none

Size	T	TT	ST	Data
0	1	1	9	

4.3.8.2 Client

If the client has already the file it MUST send this.

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_DESTRUCTION

Sub-Type: RT_SUB_FILE

Data: none

Size	T	TT	ST	Data
0	1	1	9	

if not, here is what it MUST send.

Size: Depending on the size of the data

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_CREATION

Sub-Type: RT_SUB_FILE

Data: none

Size	T	TT	ST	Data
0	1	1	9	

4.3.9 Start

When a player has his id, loaded the map and the starfield. He must send the following to say that he is ready to play.

4.3.9.1 Client

Size: 0

Type: RT_TYPE_ROOM

Transaction Type: RT_TT_REQUEST

Sub-Type: RT_SUB_START

Data: none

Size	T	TT	ST	Data
0	1	0	8	

4.3.9.2 Server

When the players are ready to play, the server MUST send this to them.

Size: 0
Type: RT_TYPE_ROOM
Transaction Type: RT_TT_ACK
Sub-Type: RT_SUB_START
Data: none

Size	T	TT	ST	Data
0	1	2	8	

4.4 Prelude

Here the server MUST notify the creation of the player's objects in the game to all the clients.
(See Object 4.5.2).

4.5 Game operations

This part will describe the several packets exchange between the client and the server during a game. Note that from now on all the packets be sent using UDP protocol. To deal with possibly lost of a packet the first 4 bytes of the data will be the timestamp of the packet.

4.5.1 Events

Each of those packets will synchronize the players events with the server. The player will tell to the server what he is doing and then the server will confirm it to him.

4.5.1.1 Movement

When the player goes up, down, left or right the client MUST send this.

4.5.1.1.1 Client

Size: 9 bytes.
Type: RT_TYPE_GAME
Transaction Type: RT_TT_MODIFICATION
Sub-Type: RT_SUB_EVENT
Data: it will contain the timestamp (4 bytes), the player's id (4 bytes), the event enum (1 byte).

Size	T	TT	ST	Data (with Timestamp)
9	2	1	0	////////////////////

4.5.1.1.2 Server

Size: 17 bytes
Type: RT_TYPE_GAME
Transaction Type: RT_TT_MODIFICATION
Sub-Type: RT_SUB_EVENT
Data: it will contain the timestamp (4 bytes), the player's id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte).

Size	T	TT	ST	Data (with Timestamp)
17	2	1	0	////////////////////////////////////

4.5.1.2 Leave

When a player leave the game, the client MUST send this. To make sure that the client will receive the packet this one will be sent by TCP over IP.

4.5.1.1.1 Client

Size: 9 bytes.
Type: RT_TYPE_GAME
Transaction Type: RT_TT_DESTRUCTION
Sub-Type: RT_SUB_EVENT
Data: it will contain the timestamp (4 bytes), the player's id (4 bytes), the event enum (1 byte).

Size	T	TT	ST	Data (with Timestamp)
9	2	1	0	////////////////////////////////////

4.5.1.1.2 Server

See 4.5.2.3 Destruction.

4.5.2 Objects

An object can be a player, a bullet, ennemy or a bonus. Here is the packet you MUST send to use them.

4.5.2.1 Creation

Size: depending on the size of the data
Type: RT_TYPE_GAME
Transaction Type: RT_TT_CREATION
Sub-Type: RT_SUB_OBJECT
Data: it will contain the timestamp (4 bytes), the id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte), filename.

Size	T	TT	ST	Data (with Timestamp)
254(bytes)max	2	0	1	/

4.5.2.2 Modification

Size: 17 bytes
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_MODIFICATION
 Sub-Type: RT_SUB_OBJECT
 Data: it will contain the timestamp (4 bytes), the id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte).

Size	T	TT	ST	Data (with Timestamp)
17	2	1	1	/

4.5.2.3 Destruction

Size: 17 bytes
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_DESTRUCTION
 Sub-Type: RT_SUB_OBJECT
 Data: it will contain the timestamp (4 bytes), the player's id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte).

Size	T	TT	ST	Data (with Timestamp)
17	2	1	2	/

4.5.3 Player's bullets

4.5.3.1 Client

Size: depending on the size of the data
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_CREATE
 Sub-Type: RT_SUB_BULLET
 Data: it will contain the timestamp (4 bytes), local object's id (4 bytes), the parent's id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte), filename.

Size	T	TT	ST	Data (with Timestamp)
254(bytes)max	2	1	2	/

4.5.3.2 Server

Size: 21 bytes
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_MODIFICATION
 Sub-Type: RT_SUB_BULLET
 Data: it will contain the timestamp (4 bytes), local object's id (4 bytes), the final's id (4 bytes), the position x (4 bytes) and y (4 bytes), direction (1 byte).

Size	T	TT	ST	Data (with Timestamp)
21	2	1	2	////////////////////////////////////

4.5.4 Scoring

The server will update the score of a player by sending this packet.

4.5.4.1 Server

Size: 8 bytes
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_MODIFICATION
 Sub-Type: RT_SUB_SCORE
 Data: ID(4 bytes) Score(4 bytes)

Size	T	TT	ST	Data
8	2	1	//	////////

4.5.5 Game Over

If everyone lost all their life, they will receive a packet notifying the end of the game. To make sure that the client will receive the packet this one will be sent by TCP over IP.

4.5.5.1 Server

Size: 0
 Type: RT_TYPE_GAME
 Transaction Type: RT_TT_DESTRUCTION
 Sub-Type: RT_SUB_GAMEOVER
 Data: none

Size	T	TT	ST	Data
0	2	2	4	

4.6 Postlude

At the end of a level the server MUST notify all the clients. To make sure that the client will receive the packet this one will be sent by TCP over IP.

4.6.1 Server

Size: 0
Type: RT_TYPE_GAME
Transaction Type: RT_TT_DESTRUCTION
Sub-Type: RT_SUB_LEVELOVER
Data: none

Size	T	TT	ST	Data
0	2	2	5	

5 TROUBLESHOOT

5.1 Latency

In each packets exchanged in the UDP protocol, a field is reserved to the timestamp to solve the problem of latency.

6 CONTACTS

Please contact us with any remarks at:

soufi en. benramdhane@epi tech. eu
alexandre. da-silva@epi tech. eu