Project Edge Network Specification

Big Documentation for a Big World

Addie Morrison

Documentation Started: September 4, 2015
Last Updated: September 13, 2015

TODO: Write an introduction or something

0.1 Key

- Implemented Feature
  - Feature to be implemented

0.2 Ports

2345	Maestro server port
2346	Maestro client port (different to avoid discovery traffic)
2347	Atlas client port
2348-2500	Atlas server ports

1 Maestro Packets

Create Lobby

Sent from a Hyperion instance to Maestro to create a new lobby

- Byte: Number of players per team
- Byte: Number of players invited
- Several Strings: Usernames of players to invite

Invite To Lobby

Sent from Maestro to Hyperion instances to invite them to a lobby

- Int32: Lobby ID
- String: Host Username

Reply To Lobby Invite

Sent from Hyperion instances to Maestro in response to an invitation

- Int32: Lobby ID
- Bool: Whether they accepted or not

Lobby Status

Sent every 2(?) seconds from Maestro to Hyperion instances in a lobby, updating them on the lobby's status

- Byte: Number of people in the lobby
- Several Strings: Usernames of people in the lobby

Start Lobby

Sent from the Hyperion instance `hosting' a lobby to Maestro telling to to place them into the queue

- Int32: Lobby ID

Introduce Atlas

Sent from Maestro to Hyperion instances giving them the information for their session

- String: Atlas host (currently defaults to same address as Maestro)
- Int32: Atlas port number
- String: Session encryption key

2 Atlas Packets

Debugging Structures

Request Position Change

Hyperion requests a position change

- UInt16: X position
- UInt16: Y position

Update Positions

Atlas tells Hyperion where each player is

- UInt16: Number of players being described in this message
- Several player data 'Structures'
  - Int64: Net Remote UUID
  - UInt16: Player X position
  - Uint16: Player Y position

Actual Packets

(Waiting on a class hierarchy to be started before attempting to transmit data about it)