

James Odebiyi

Yann Nkabong

Simi Olusa

# **RCST Protocol**

November 2022

## **Status of this Memo**

This document specifies an application protocol used to manage data transfer between a controller, a renderer and a server. Please refer to this document for questions about the application protocol.

## **Abstract**

The Receiver Controller Server Transfer (RCST) protocol, is a text based application-level protocol that specifies the methods in which a controller ( C ) can request a list of media files located on a server ( S ) and then request a renderer ( R ) to request the server (S) to stream a specific media file to the renderer for rendering.

Table of contents		Page
1	Introduction	<a href="#"><u>3</u></a>
2	NodeAdresses	<a href="#"><u>3</u></a>
2.1	controllerIP	<a href="#"><u>3</u></a>
2.2	serverIP	<a href="#"><u>3</u></a>
2.3	rendererIP	<a href="#"><u>3</u></a>
3	NodePorts	<a href="#"><u>4</u></a>
3.1	serverPort	<a href="#"><u>4</u></a>
3.2	rendererPort	<a href="#"><u>4</u></a>
3.3	controllerPort	<a href="#"><u>4</u></a>
4	MessageType	<a href="#"><u>4</u></a>
4.1	RETRIEVE	<a href="#"><u>5</u></a>
4.2	RENDER	<a href="#"><u>5</u></a>
4.3	STREAM	<a href="#"><u>5</u></a>
4.4	CLOSE	<a href="#"><u>6</u></a>
4.5	RESPONSE	<a href="#"><u>6</u></a>

# **1 – Introduction**

The purpose of this protocol is to provide an efficient and reliable way of communicating messages between a controller, renderer and server. This specification document outlines the various message types used in the protocol, their functions as well as the overall structure of an RCST header.

## **2 – NodeAddresses**

RCST protocol defines the addresses of all 3 devices used in the application: controller, server, and renderer. The general syntax goes as follows: “xx.x.x.x”.

### **2.1 – controllerIP**

The controller IP has an address of “10.0.0.1”. The IP is defined and will be used by the other 2 devices to send messages through the network.

### **2.2 – serverIP**

The server IP has an address of “10.0.0.2”. The IP is defined and will be used by the other 2 devices to send messages through the network.

### **2.3 – rendererIP**

The renderer IP has an address of “10.0.0.3”. The IP is defined and will be used by the other 2 devices to send messages through the network.

## **3 – NodePorts**

RCST protocol defines the ports of all 3 devices used in the application: controller, server, and renderer. The general syntax goes as follows: “xxxx”.

### **3.1 – serverPort**

The server has a port number of 5000. The port number is defined and will be used by the other 2 devices to determine where the message is will delivered exactly.

### **3.2 – rendererPort**

The renderer has a port number of 3000. The port number is defined and will be used by the other 2 devices to determine where the message is will delivered exactly.

### **3.3 – controllerPort**

The controller has a port number of 2000. The port number is defined and will be used by the other 2 devices to determine where the message is will delivered exactly.

## **4 – MessageType**

RCST protocol defines 5 different message types that work together to implement the application: RETRIEVE, RENDER, STREAM, CLOSE, and RESPONSE. The general syntax goes as follows: {“MessageType”: <message type>, “ResourceName”: <text, media, ...>,

“Source IP”: <IP number>, “Render IP”: <IP number>, “Source Port”: <port number>,  
“Destination Port”: <port number>}.

## **4.1 – RETRIEVE**

Retrieve syntax goes as follows: {“MessageType”: <RETRIEVE>, “ResourceName”:  
<text, media, ...>, “Server IP”: <IP number>, “Version”: <1.0>}

The controller sends a message that contains the type of to be sent, the resource name it wishes to retrieve the data from, the server IP, and the version of the data.

## **4.2 - RENDER**

Render syntax goes as follows: {“MessageType”: <RENDER>, “ResourceName”: <  
{“resourcePath”: <string>, “resource”: <string>} >, “Server IP”: <IP number>, “Version”:  
<1.0>}

The controller sends a message that contains the type of message sent, the resource name (containing an object composed with the resource path and the resource itself) it wishes to retrieve the data from, the server IP, and the version of the data.

## **4.3 - STREAM**

Stream syntax goes as follows: {“MessageType”: <STREAM>, “Server IP”: <IP  
number>, “Version”: <1.0>, “Status”: <RESPONSE>, “payload”: <string>}

The renderer sends a message that contains the type of message sent, the server IP, the version, the status, and the payload of the data.

## 4.4 - CLOSE

Close syntax goes as follows: {"MessageType": <CLOSE>, "ResourceName": <Server>, "ServerIP": <IP number>, "Version": <1.0>}

The controller sends a message that contains the type of message sent, resource name of the element it wishes to close, the serverIP and the version of the data.

## 4.5 - RESPONSE

Response syntax goes as follows: {"MessageType": <RESPONSE>, "Server IP": <IP number>, "Version": <1.0>, "Status": <RESPONSE>, "payload": <string>}

The render sends a message that contains the type of message sent, the server IP, the version, the status, and the payload of the data.