



流媒体：RTMP 命令消息与流管理



小岛上的...

深度智障 理工双学位 - 视觉黑桃 菜鸟炼丹师

1 人赞同了该文章

• 背景

上文中我们详细介绍了 RTMP (Real Time Messaging Protocol) 的规格及关键组成部分，这一章我们来讲解 RTMP 怎样通过命令消息，来进行实际的C-S交互控制。需要注意的是，命令消息的格式依然和上文中的统一格式一致。这里的区别主要是指携带数据（即消息体）的不同。

一、命令消息

C-S端用AMF编码后的命令消息，来进行对整个 RTMP 流程的控制。命令消息是链接成功建立后，推拉流及相关操作的关键控制手段。因此，通常会根据所处理业务，携带相应的关键数据。同时由于是对流程的管理手段，相应命令往往也需要对端对结果进行回应。这类回应消息，通过 Transaction ID 来标记被回应的方法，并通过固定的命名格式来进行约束：

[命令消息] + _result 或者 [命令消息] + _error

命令消息主要分为两种类型：

1. **NetConnection**：网络连接命令，用来管理 C-S 间双向链接，并提供远程方法异步调用。
2. **NetStream**：数据流命令，被用来管理 音视频流 和 数据通道 相关的管理和业务操作。

接下来我们来分别看一下这两者。

二、网络连接命令 NetConnection

网络连接命令，主要有方法如下：

| Command Name | 作用 |
|---------------|------------------------------|
| connect | C端发送给S端，以请求建立S端连接应用实例 |
| call | 请求对端（receiver）执行RPC服务，携带RPC名 |
| close | C端发送给S端，以请求释放S端连接应用实例 |
| createStream | C端发送给S端，以请求S端为音视频数据建立通信信道 |
| releaseStream | C端发送给S端，通知S端释放数据信道（C端推流时） |

connect 指令

connect 指令是由客户端发给服务器端，用来获取一个可用于建立对端通信的服务端应用实例的一个指令。指令分为请求指令 connect 和 应答指令 connect_result。

请求指令主要由4部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|-------------------------|-----------------|--------------|
| Command Name | AMF-String | "connect" |
| Transaction id | AMF-Number | 总是 1 |
| Command Object | AMF-Object（Map） | 连接配置参数名-值对列表 |
| Optional User Arguments | AMF-Object（Map） | 额外配置信息（双端协定） |

应答指令主要由4部分数据组成：

| | | |
|----------------|------------------|--|
| Transaction id | AMF-Number | 总是 1 |
| Properties | AMF-Object (Map) | 连接确认的配置属性名-值对列表 |
| Information | AMF-Object (Map) | 描述应答特征的信息，携带如：'code', 'level', 'description' 之类的字段 |

对于 connect 请求/应答指令，可用于连接配置的参数如下：

| 属性 | 数据类型 | 作用 | 示例 |
|----------------|-------------|---|---|
| app | AMF-String | 想要链接的服务端应用名称 | testapp |
| flashver | AMF-String | Flash Player 的版本号 | FMS/1.0 |
| swfUrl | AMF-String | SWF 源文件的连接地址 URL | file://c:/t1.swf |
| tcUrl | AMF-String | 服务端协议实例 URL，格式规定为： protocol://server:port/app/instance | rtmp://localhost:1935/ testapp/instance1 |
| fpad | AMF-Boolean | 如果使用代理（proxy）则为 True | true/false |
| audioCodecs | AMF-Number | 表明客户端支持的音频解码类型 | SUPPORT_SND _MP3 |
| videoCodecs | AMF-Number | 表明客户端支持的视频解码类型 | SUPPORT_VID _SOBRENSON |
| videoFunction | AMF-Number | 表明客户端支持的视频操作 | SUPPORT_VID _CLIENT_SEEK |
| pageUrl | AMF-String | SWF 源文件的来源页面 URL | http://ahost/a.html |
| objectEncoding | AMF-Number | AMF 编码格式 | AMF3 |

显而易见，可配置参数里，音视频编解码格式及一些规格化的属性，是由 RTMP 规范了配置标准的。主要是对 audioCodecs、videoCodecs、videoFunction、objectionEncoding进行了规范。我们依次来看一下：

| | | |
|---------------------|--------|--------------------------------|
| SUPPORT_SND_ADPCM | 0x0002 | ADPCM 压缩格式 |
| SUPPORT_SND_MP3 | 0x0004 | mp3 压缩格式 |
| SUPPORT_SND_INTEL | 0x0008 | intel 的压缩格式，已经被淘汰了 |
| SUPPORT_SND_UNUSED | 0x0010 | 没用/不支持 |
| SUPPORT_SND_NELLY8 | 0x0020 | NellyMoser 8-kHz 压缩格式 |
| SUPPORT_SND_NELLY | 0x0040 | NellyMoser 5/11/22/44-kHz 压缩格式 |
| SUPPORT_SND_G711A | 0x0080 | Flash 媒体服务器专用 G711A 格式 |
| SUPPORT_SND_G711U | 0x0100 | Flash 媒体服务器专用 G711U 格式 |
| SUPPORT_SND_NELLY16 | 0x0200 | NellyMoser 16-kHz 压缩格式 |
| SUPPORT_SND_AAC | 0x0400 | AAC (Advanced audio coding) 格式 |
| SUPPORT_SND_SPEEX | 0x0800 | Speex Audio 压缩格式 |
| SUPPORT_SND_ALL | 0x0FFF | 上面的所有类型都支持 |

| Video Codec Flag | 取值 | 代表类型 |
|-------------------------|--------|---------------------------------|
| SUPPORT_VID_UNUSED | 0x0001 | 没用/不支持 |
| SUPPORT_VID_JPEG | 0x0002 | 废弃不用 |
| SUPPORT_VID_SORENSEN | 0x0004 | Sorenson Flash video 格式 |
| SUPPORT_VID_HOMEBREW | 0x0008 | V1 Homebrew 屏幕共享 (需要 Flash 8+) |
| SUPPORT_VID_VP6 (On2) | 0x0010 | On2 视频 (需要 Flash 8+) |
| SUPPORT_VID_VP6ALPHA | 0x0020 | On2 + alpha通道 的视频 (需要 Flash 8+) |
| SUPPORT_VID_HOMEBREWV | 0x0040 | V2 Homebrew 屏幕共享 (需要 Flash 8+) |
| SUPPORT_VID_H264 | 0x0080 | H.264 视频格式 |
| SUPPORT_VID_ALL | 0x00FF | 上面的所有类型都支持 |

| Video Function Flag | 取值 | 代表类型 |
|-------------------------|--------|---------------|
| SUPPORT_VID_CLIENT_SEEK | 0x0001 | 表示客户端支持精准Seek |

| Object Encoding Type | 取值 | 代表类型 |
|----------------------|----|-----------------------|
| AMF0 | 1 | 采用 AMF0 编码 (Flash 6+) |

call 指令

指令发送端通过call 指令，来调用对端的 Remote Procedure Calls（即远程程序调用 RPC），来执行对端指定的方法或处理过程，获取结果返回。具体程序需要独立实现。指令分为请求指令 call 和 应答指令 call_result。

请求指令主要由4部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|-------------------------|-----------------|--------------------------|
| Procedure Name | AMF-String | 指定调用对端的远程功能名称 |
| Transaction id | AMF-Number | 事务 ID，用来标记请求，如不需要也可传0 |
| Command Object | AMF-Object（Map） | RPC 需求的配置参数名-值对列表，无则null |
| Optional User Arguments | AMF-Object（Map） | 额外配置信息（双端协定） |

应答指令主要由4部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|--------------------------|
| Command Name | AMF-String | RPC 请求方定义的回应方法的名称 |
| Transaction id | AMF-Number | 需要响应的 RPC 请求的事物 ID |
| Command Object | AMF-Object（Map） | RPC 需求的配置参数名-值对列表，无则null |
| Response | AMF-Object（Map） | 被调用的 RPC 方法返回结果 |

close 指令

close 指令在官方的 RTMP 规格说明文档中并未公开，如用到需要双端协定。（这个指令貌似是专门针对 Flash Media Player 流媒体处理上的一则资源协同释放的指令，如果不用 Flash Media Player 的话，貌似并不需要用到）

createStream 指令

createStream 指令是由C端发送给S端，来通知服务端建立一个能够进行音视频数据及相关元数据传输的流媒体信道。服务端在收到客户端的 createStream 请求，会在信道创建成功后，向客户端发送应答指令，告知客户端当前用来做数据传输的 ms_id（Message Stream ID）。指令分为请求

请求指令主要由3部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|--------------------------|
| Command Name | AMF-String | “createStream” |
| Transaction id | AMF-Number | 事务 ID，用来标记请求 |
| Command Object | AMF-Object（Map） | RPC 需求的配置参数名-值对列表，无则null |

应答指令主要由4部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|--|
| Command Name | AMF-String | “createStream_result”/“createStream_error” |
| Transaction id | AMF-Number | 对应请求“createStream”的事物 ID |
| Command Object | AMF-Object（Map） | RPC 需求的配置参数名-值对列表，无则null |
| Stream ID | AMF-Number | 数据信道 ms_id |

releaseStream 指令

releaseStream 指令是由C端发送给S端，在C端作为推流方，用来在推流结束时通知服务端销毁当前数据信道分配的资源 的命令。分请求指令 releaseStream 和 应答指令 releaseStream_result。

请求指令主要由4部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|-----------------|
| Command Name | AMF-String | “releaseStream” |
| Transaction id | AMF-Number | 事务 ID，用来标记请求 |
| Command Object | AMF-Object（Map） | 没有命令参数，null |
| Stream Name | AMF-String | 要终止的推流数据流命名 |

应答指令主要由3部分数据组成：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|------------|--|
| Command Name | AMF-String | “releaseStream_result”/“releaseStream_error” |
| Transaction id | AMF-Number | 对应请求“releaseStream”的事物 ID |

数据流命令是在数据信道建立完毕后，用来对音视频数据流进行直接控制的命令类型。这些命令作用于当前信道对应数据的操控行为。常用的命令有以下几个：

| Command Name | 作用 |
|--------------|---------------------------------------|
| play | C端发送给S端，以请求建立S端开始数据播放（传输） |
| play2 | C端发送给S端，同 play，但可动态切换bitrate而不改变已播放数据 |
| deleteStream | C端发送给S端，以请求S端销毁当前 ms_id 对应的数据信道 |
| closeStream | C端发送给S端，以请求S端关闭当前 ms_id 对应的数据信道 |
| receiveAudio | C端发送给S端，通知发送音频数据时机 |
| receiveVideo | C端发送给S端，通知发送视频数据时机 |
| publish | C端发送给S端，通知S端准备接收推流数据（C端推流时） |
| seek | C端发送给S端，通知S端跳转到指定时间戳位置，后续从此开始 |
| pause | C端发送给S端，通知S端暂停数据推送（C端拉流时） |

数据流命令的请求命令所对应的应答命令，被统一命名为 “onStatus” 以描述数据所处的状态发生变更，因此具有统一的格式：

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|---|
| Command Name | AMF-String | “onStatus” |
| Transaction id | AMF-Number | 恒为 0 |
| Command Object | AMF-Null | 状态应答中不包含命令配置数据 |
| Info Object | AMF-Object（Map） | 一系列包含有当前状态详情及相关数据的 AMF 编码信息。至少含有： 1. 'level': 级别（warning、status、error） 2. 'code': 触发源，如 NetStream.Play.Start 3. 'description': 描述，自定义附加解释说明 其余额外的信息，由双端自行协定 |

因此，我们通过处理 NetStream 的应答指令的 ‘level’ 和 ‘code’ 字段，就可以知道相应的请求指令是否有成功执行了。进而判断当前状态，并确认是否需要执行后续的操作流程。

这一部分的命令就很明显存在制式化。因此就直接列出参数说明了。

当 error 发生时，除了 onStatus 异常外，还会返回 [方法名]_error 消息返回，这个也是一致的。

| | | |
|----------------|-------------|---|
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数，null |
| Stream Name | AMF-String | 被播放的流的源文件格式与命名。 FLV 的播放，在这里有命名规范，我们需要将源文件和其格式，按照： [格式]:[文件名（携带扩展名）] 的形式统一命名。例如： “mp3:song1”、“mp4:sample.m4v” |
| Start | AMF-Number | 可选参数 指定开始播放的起始时间位置。默认 -2。 -2 表示：先尝试本地缓存，无则拉流； -1 表示：只使用本地缓存 ≥0 表示：从指定位置播放，先缓存，无则拉流，如果找不到对应 Stream Name 的文件，则尝试遍历播放列表（playlist）下一位。 |
| Duration | AMF-Number | 可选参数 指定播放持续时长。默认 -1。 -1 表示：持续播放，直到流或视频终结； 0 表示：只播放一帧 >0 表示：视频播放定义时长，超时则停止 |
| Reset | AMF-Boolean | 参数决定是否需要刷新掉之前的播放列表（playlist）的缓冲数据 |

play2 指令

| name（按序取 填） | 数据类型 | 作用 |
|----------------|-----------------|--|
| Command Name | AMF-String | “play2” |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数，null |
| Parameters | AMF-Object（Map） | 一系列包含有 Flash 播放配置相关共有属性，这些信息可以参见 flash.net.NetStreamPlayOptions ActionScript 里的详细说明。信息采用 AMF 编码。 这些配置被用来做播放的动态参数变更 NetStreamPlayOptions 对象的公开属性在 ActionScript 3 语言指南中 [AS3] 有所描述。 |

deleteStream 指令

| | | |
|-------------------|------------|----------------|
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数, null |
| Message Stream ID | AMF-Number | 已被销毁的数据信道对应 ID |

注意：这个消息只用来通知服务器，没有统一应答返回。

closeStream 指令

| name (按序取 填) | 数据类型 | 作用 |
|----------------|------------|---------------|
| Command Name | AMF-String | "closeStream" |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数, null |

receiveAudio 指令

| name (按序取 填) | 数据类型 | 作用 |
|----------------|-------------|-----------------|
| Command Name | AMF-String | "receiveAudio" |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数, null |
| Bool Flag | AMF-Boolean | 是否需要服务端发送新的音频数据 |

注意：
如果，Bool Flag = false 这个消息只用来通知服务器，没有统一应答返回。
如果，Bool Flag = true则应答消息' code' 为 NetStream.Seek.Notify 或 NetStream.Play.Start

receiveVideo 指令

| name (按序取 填) | 数据类型 | 作用 |
|----------------|------------|----------------|
| Command Name | AMF-String | "receiveVideo" |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数, null |

如果，Bool Flag = true则应答消息‘code’为 NetStream.Seek.Notify 或 NetStream.Play.Start

publish 指令

| name (按序取 填) | 数据类型 | 作用 |
|-----------------|------------|---|
| Command Name | AMF-String | “publish” |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数，null |
| Publishing Name | AMF-String | 发布的数据流的命名 |
| Publishing Type | AMF-String | 发布数据流的类型，有下面3种： 1. ‘live’: 标准直播，不保存数据为文档 2. ‘record’: 录制，要求S端进行录制存根 3. ‘append’: 续播，接续之前同名文件记录 |

注意：

publish 的返回状态 NetStream.Publish.Start，这个消息不止由 onStatus 统一应答携带，也会由 onFCPublish 返回。

其实RTMP 是有 unpublish 方法的，相应触发状态为 NetStream.Unpublish.Success，同样也会由除 onStatus 外的 onFCUnpublish 返回。不过这个方法通常都被我们以 releaseStream 直接涵盖过去了。且 RTMP 公开的规格文档里并没有描述，因此一般不用。

seek 指令

| name (按序取 填) | 数据类型 | 作用 |
|----------------|------------|-------------|
| Command Name | AMF-String | “seek” |
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数，null |
| milliseconds | AMF-Number | 跳转位置时间戳 |

注意：seek 成功后，会有 ‘code’ 为 NetStream.Seek.Notify 消息返回。

pause 指令

| | | |
|--------------------|-------------|--|
| Transaction id | AMF-Number | 总是 0 |
| Command Object | AMF-Null | 没有命令参数，null |
| Pause/Unpause Flag | AMF-Boolean | 是否分别代表暂停和恢复 |
| milliSeconds | AMF-Number | 暂停时，记录流被暂停时的时间位置 恢复时，表示需要携带上一次暂停的时间位置 |

注意：

pause 成功后，会有 'code' 为 NetStream.Pause.Notify 消息返回。

unpause 成功后，会有 'code' 为 NetStream.Unpause.Notify 消息返回。

四、总结

至此，RTMP 命令消息的详细规格说明，以及 RTMP 连接过程的命令控制，就讲解完毕了。结合上一篇：《流媒体：RTMP 协议全面解析》，我们基本上梳理了整个 RTMP 的全部关键节点。剩下的就需要实操实践啦～

希望能对你有所帮助～

参考文献：

[1] RTMP 官方规范 (<https://www.adobe.com/devnet/rtmp.html>)

发布于 2020-08-24

流媒体 直播 rtmp

文章被以下专栏收录



音视频研发与AI视觉

音视频技术积累资源池，个人经验总结

进入专栏



流媒体：RTMP 协议完全解析

小岛上的黑... 发表于音视频研发...



流媒体传输协议：RTMP、和RTSP介绍

田野

还没有评论

写下你的评论...

