# red5视频直播

## red5

red5是一个由java编写的开源flash服务器。是建立在Jetty(servlet engine) ， Mina (networking)基础之上的，参考了Terracotta Clustering技术理念，并通过使用Spring框架将起整合起来的。它支持：

* Streaming Video (FLV, F4V, MP4, 3GP)
* Streaming Audio (MP3, F4A, M4A, AAC)
* Recording Client Streams (FLV and AVC+AAC in FLV container)
* Shared Objects
* Live Stream Publishing
* Remoting
* Protocols: RTMP, RTMPT(包装在http中穿越防火墙，作为1935被封时的rtmp备选), RTMPS(HTTPS连接), and RTMPE(加密)

可以把音频和视频转换成播放流； 录制客户端播放流（只支持FLV）；共享对象；现场直播流发布；远程调用(Remoting)。red5使用rtmp作为流媒体传输协议，在其自带的一些示例中演示了在线录制，flash流媒体播放，在线聊天，视频会议等一些基本功能。使用Shared Objects可以做网页游戏的信息同步分发，但性能欠佳，应用不多。

## rtmp

RTMP（Real Time Messaging Protocol）是Adobe公司开发的专利协议，被 Flash 用于对象、视频、音频的传输。该协议建立在 TCP 协议或者轮询 HTTP 协议之上， RTMP 协议就像一个用来装数据包的容器，这些数据可以是 AMF 格式的数据，也可以是 FLV 中的视 / 音频数据。一个单一的连接可以通过不同的通道传输多路网络流，这些通道中的包都是按照固定大小的包传输的。

## 直播发布



一 连接服务器：

通过rtmp协议，从发布客户端创建一个连接。当网络状态为“NetConnection.Connect.Success”表示连接建立成功。此时，初始化摄像头和麦克，获取视频、声音输入。创建一个放置在网络连接上的流，将视频声音附着在该流上，以固定大小的包通过网络连接与red5服务器通信。



附着着视频和声音的流被发送到服务器端，在服务器端该流可以进行直播发布、录制、追加录制以及附带信息的追加录制。

二 参数设置：

流畅，是直播的一个重要直观感受。流畅与否主要取决于画面品质、设置的可用带宽以及捕捉频率，其中前两者需要配合设置。

设置了一个画面品质数值和可用带宽数值后，当带宽无法满足此品质的视频传输时，就会出现“卡”的现象。品质越低需要的带宽越小。

当传输摄像头所捕捉的视频帧过多导致传输内容过多时，也会出现“卡”的现象，可以通过减少传输视频帧的办法解决，但画面会连贯性相应降低。

声音相对视频对流畅程度影响较小。

最后，需要注意设备的使用条件。摄像头麦克风可以满足视频会议的需求。当使用如摄像机之类相对高端的设备时，需要采集卡将其与PC连接。若需要接收其他设备的模拟信号时，还需要声卡转换为数字信号（此情况常见于声音输入，解决防风、回音等情况）。

## 直播接收



一 连接服务器：

创建连接，与发布端连接相同red5服务器。在该连接上建立流，播放发布端捕获的视频声音内容（播放与发布的名称保持一致）。当接收端网络防火墙对rtmp协议端口禁用时，可以使用rtmpt方式解决。

二 直播前后：

通常在直播开始和结束后，会播放一些宣传片或者广告，同时播放端会轮询服务器检测是否有直播信号。（当直播并录制的时候，已录制的内容在直播中断时会让播放端无法区分，所以不能一直连接，当发现有已录制的flv文件存在时需要停止连接）

三 直播中断

直播过程中，发布端中断发布，接收端画面停止，由于连接并未中断，当直播恢复时接收端继续直播。在以上过程中，接收端有相应事件发送，可以用作提示和中断、重新连接等操作。