Захват RTMP-потока, ретранслируемого с другого сервера

- Описание
- Настройка AMS для републикации потока
- Тестирование

Описание

WCS может захватывать RTMP-поток, переданный другим сервером. Технически, захват ретранслируемого потока для WCS-сервера не отличается от публикации потока при помощи RTMP-кодировщика или Flash-приложения. Рассмотрим Adobe Media Server в качестве источника RTMP-потока для WCS.

Настройка AMS для републикации потока

Adobe Media Server - это серверная платформа для стриминга видео, ориентированная на клиентов, использующих Adobe Flash Player. По умолчанию, сервер предназначен для публикации потока, поэтому для републикации необходимо создать отдельное приложение.

1.Допустим, AMS установлен на Linux-сервере в каталоге /opt/adobe/ams. Приложения сервера располагаются в подкаталоге applications. Создаем каталог приложения republish

```
cd /opt/adobe/ams/applications
mkdir republish
```

2. В каталоге/opt/adobe/ams/applications/republish создаем скрипт приложения main.asc

Настройка переменных скрипта:

```
var wcsServer = "192.168.0.5";
var netConnections = new Object();
var streams = new Object();
var roomName = "#amsroom1";
```

Здесь

- wcsServer - адрес WCS сервера для републикации;
- roomName - суффикс для добавления к имени потока на WCS сервере.

Обработка присоединения клиента для публикации потока. Здесь устанавливается соединение с WCS сервером для републикации:

```
application.onConnect = function (client){
    trace("onConnect "+client.id);
    var nc = new NetConnection();
    nc.ping = function(){
        nc.call("pong",null);
    }
    nc.connect("rtmp://"+wcsServer+":1935/live");
    nc.onStatus = function(info){
        trace("onStatus info.code: "+info.code);
        if (info.code=="NetConnection.Connect.Success"){
            trace("connection opened: "+wcsServer);
        }
    }
    netConnections[client.id]=nc;
    trace("onConnect done");
    return true;
}
```

Обработка публикации потока клиентом. Здесь опубликованный поток ретранслируется на WCS сервер, при этом к имени потока добавляется суффикс:

```
application.onPublish = function(client, myStream){
    var wcsStreamName = myStream.name+roomName;
    trace("onPublish "+myStream.name+" by client.id "+client.id);
    var nc = netConnections[client.id];
    var ns = new NetStream(nc);
    ns.onStatus = function(info){
        if (info.code == "NetStream.Publish.Start"){
            trace("now publishing "+myStream.name);
        }
    }
    ns.attach(myStream);
    ns.publish(wcsStreamName);
    streams[myStream.name]=ns;
    trace("published stream "+wcsStreamName+" to: "+wcsServer);
    ns.publish(false);
    ns.publish(wcsStreamName);
}
```

Обработка завершения публикации потока клиентом. Здесь останавливается републикация на WCS сервер:

```
application.onUnpublish = function(client, myStream){
    trace("onUnpublish "+myStream.name+" by client.id "+client.id);
    var ns = streams[myStream.name];
    if (ns){
        ns.publish(false);
        var s = Stream.get(myStream.name);
        Stream.destroy(s);
        delete streams[myStream.name];
        trace("unpublished "+myStream.name);
    }
}
```

Обработка закрытия соединения клиентом. Здесь закрывается соединение с WCS сервером:

```
application.onDisconnect = function (client){
    trace("onDisconnect "+client.id);
    var nc = netConnections[client.id];
    if (nc){
        nc.close();
        delete netConnections[client.id];
        trace("disconnected "+client.id);
    }
}
```

Скрипт приложения republish

```
var wcsServer = "192.168.0.5";
var netConnections = new Object();
var streams = new Object();
var roomName = "#amsroom1";

application.onConnect = function (client){
    trace("onConnect "+client.id);
    var nc = new NetConnection();
    nc.ping = function(){
        nc.call("pong",null);
    }
    nc.connect("rtmp://"+wcsServer+":1935/live");
    nc.onStatus = function(info){
        trace("onStatus info.code: "+info.code);
        if (info.code=="NetConnection.Connect.Success"){
            trace("connection opened: "+wcsServer);
        }
    }
    netConnections[client.id]=nc;
    trace("onConnect done");
    return true;
}

application.onDisconnect = function (client){
    trace("onDisconnect "+client.id);
    var nc = netConnections[client.id];
    if (nc){
        nc.close();
        delete netConnections[client.id];
        trace("disconnected "+client.id);
    }
}

application.onPublish = function(client, myStream){
    var wcsStreamName = myStream.name+roomName;
    trace("onPublish "+myStream.name+" by client.id "+client.id);
    var nc = netConnections[client.id];
    var ns = new NetStream(nc);
    ns.onStatus = function(info){
        if (info.code == "NetStream.Publish.Start"){
            trace("now publishing "+myStream.name);
        }
    }
    ns.attach(myStream);
    ns.publish(wcsStreamName);
    streams[myStream.name]=ns;
    trace("published stream "+wcsStreamName+" to: "+wcsServer);
    ns.publish(false);
    ns.publish(wcsStreamName);
}

application.onUnpublish = function(client, myStream){
    trace("onUnpublish "+myStream.name+" by client.id "+client.id);
    var ns = streams[myStream.name];
    if (ns){
        ns.publish(false);
        var s = Stream.get(myStream.name);
        Stream.destroy(s);
        delete streams[myStream.name];
        trace("unpublished "+myStream.name);
    }
}
```

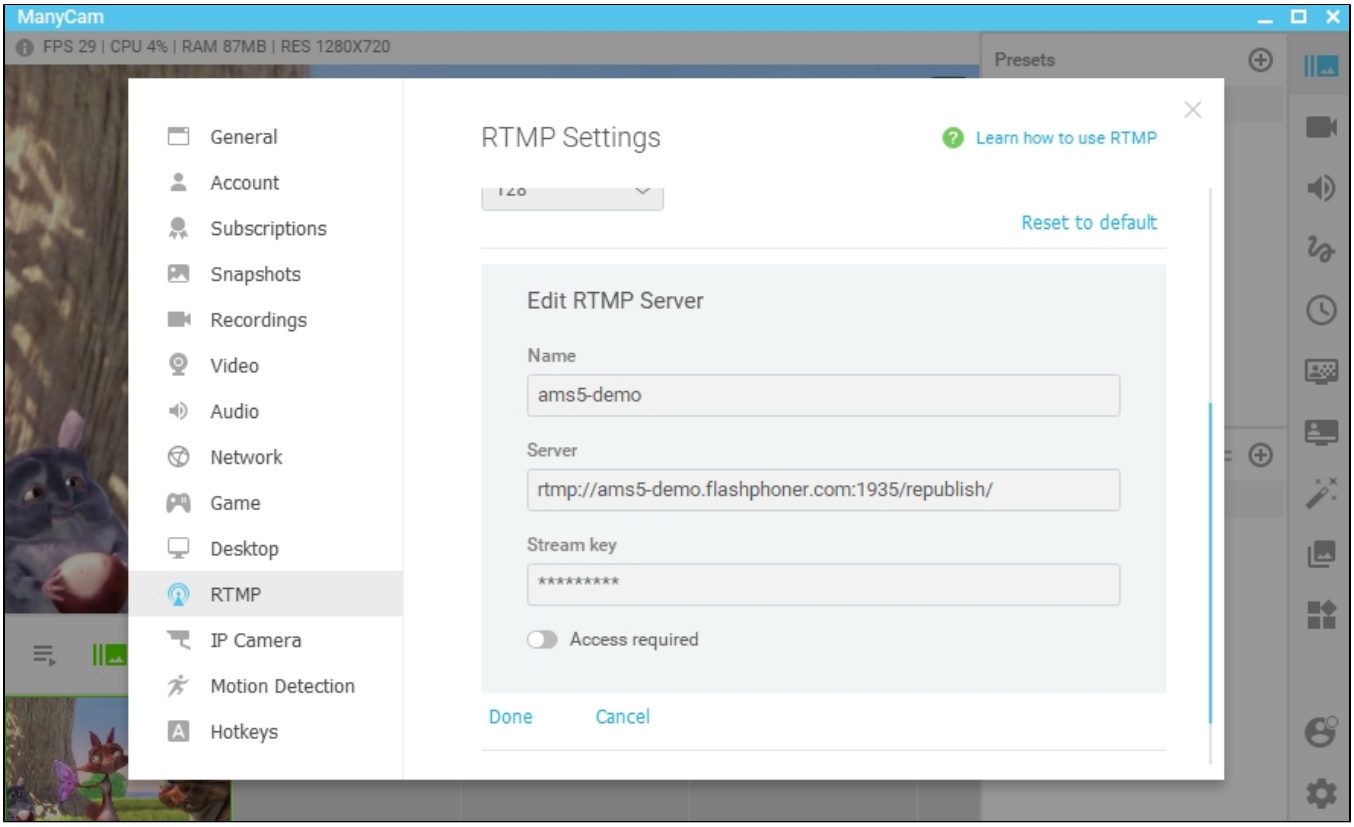
Приложение будет доступно на AMS по адресу rtmp://youramsserver:1935/republish, гдеyouramsserver - адрес Вашего AMS-сервера.

Тестирование

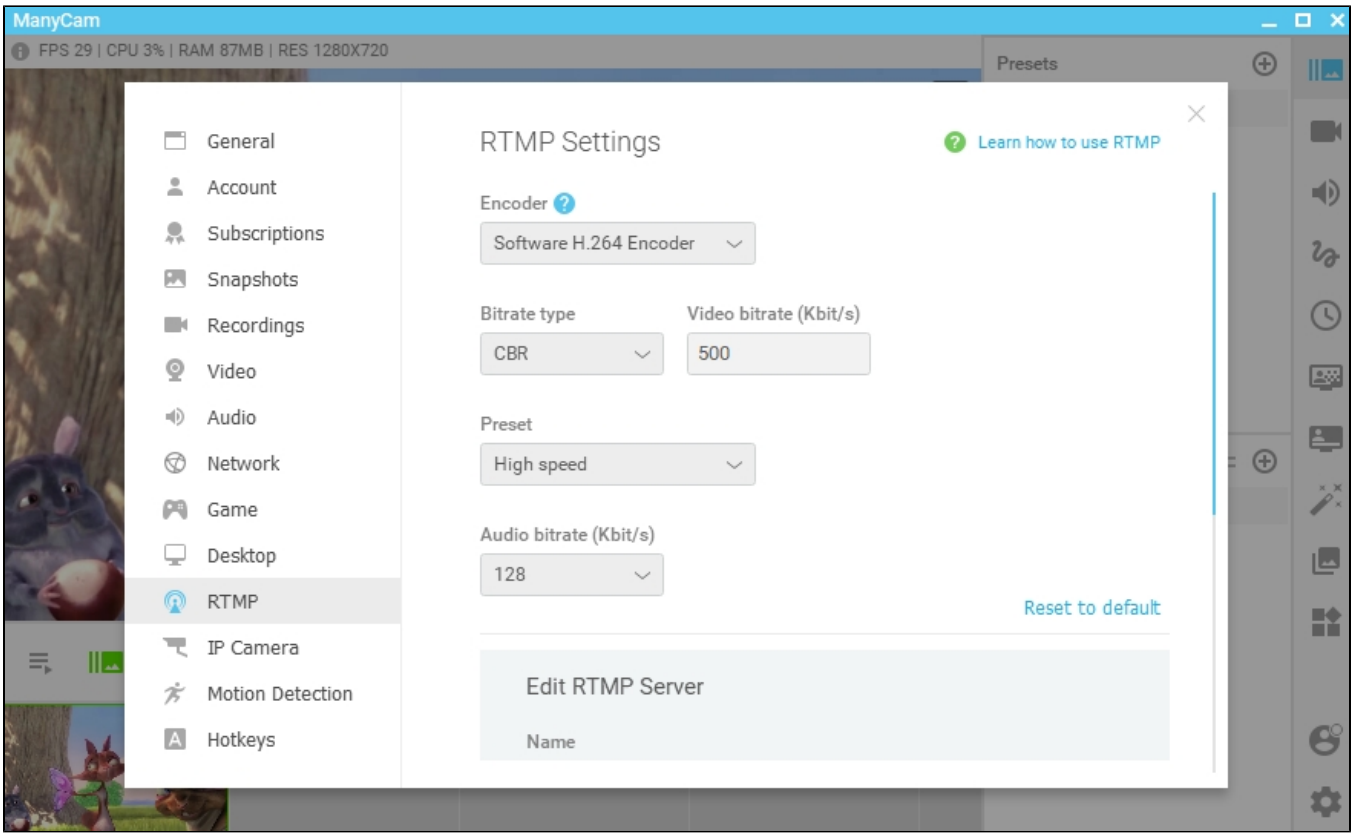
1. Для теста используем:

- AMS сервер ams5-demo.flashphoner.com
- WCS сервер mixer-demo.flashphoner.com
- ManyCam Virtual webcam для публикации RTMP-потока
- веб-приложение Playerдля воспроизведения потока с WCS сервера

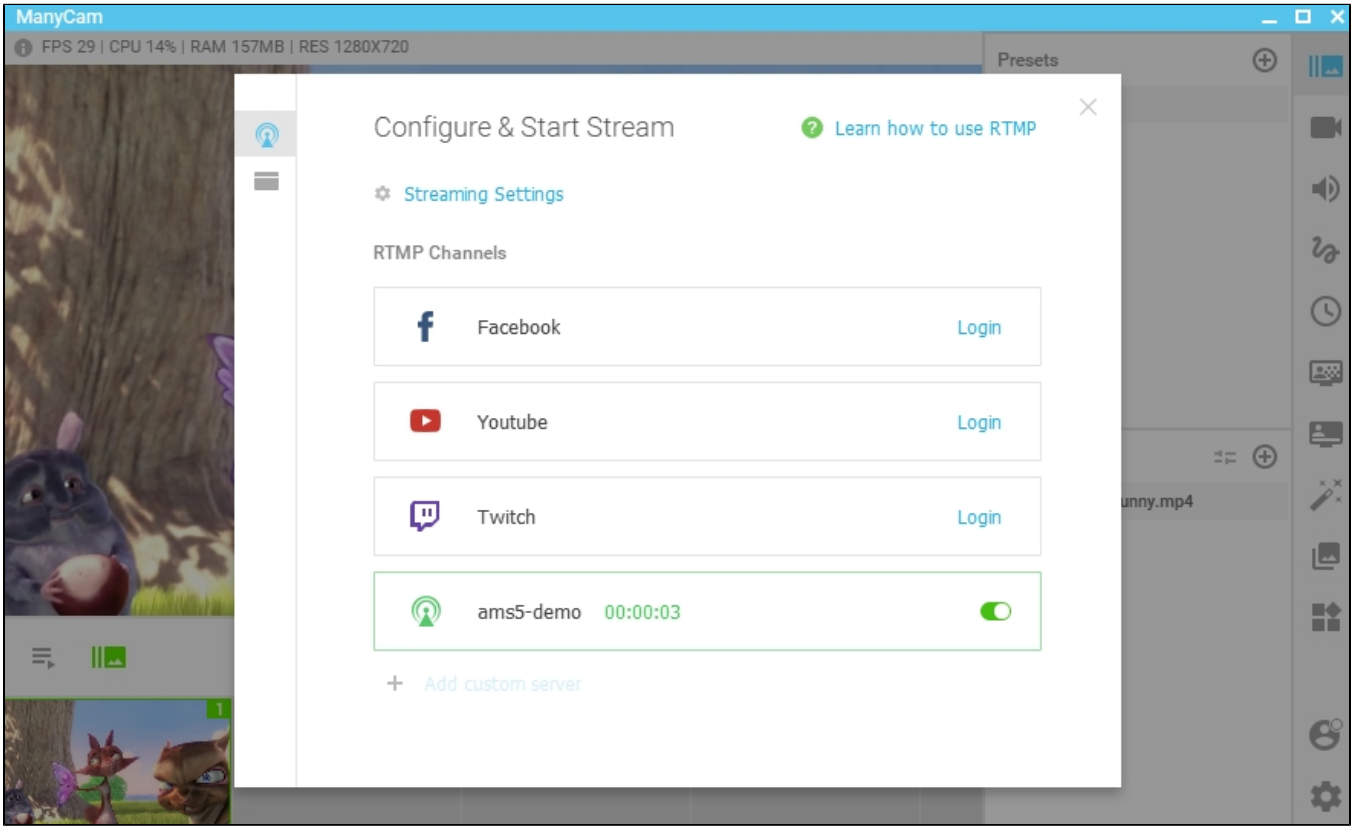
2. Настройте вещание RTMP потока в ManyCam на AMS, имя потока amsStream

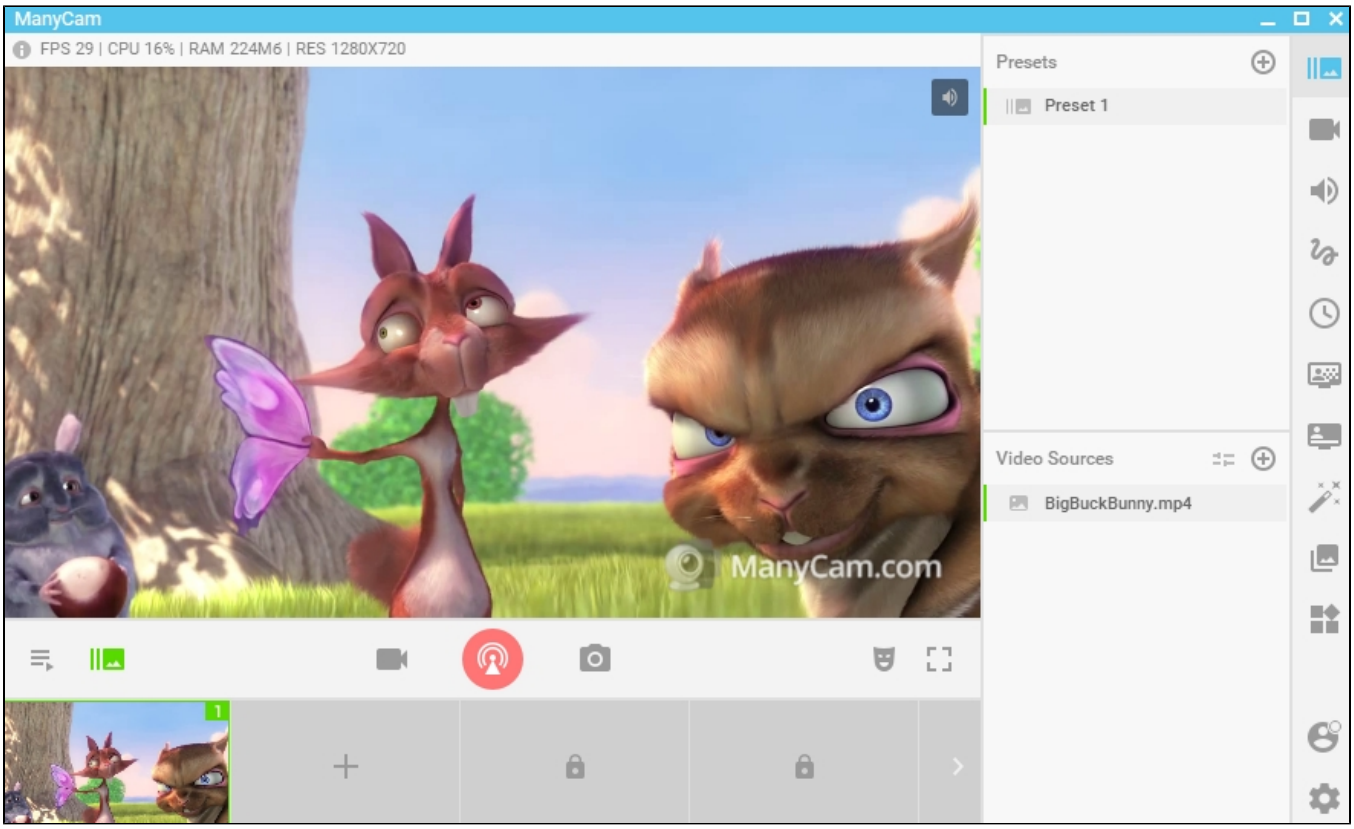


3. Настройте параметры потока в ManyCam и нажмите 'Done'



4. Запустите вещание в ManyCam





5. Откройте веб-приложение Player на WCS сервере. Укажите имя потока amsStream#amsroom1 в поле 'Stream' и нажмите 'Start'. Начнется воспроизведение захваченного потока

Player

WCS URL	wss://mixer-demo.flashphoner.co
Stream	amsStream#amsroom1
Volume	