**MDC 通信协议**

**Mobile Data Connection**

# 概述

适合于广域网、移动手机的通信协议。是基于TCP/IP的长链接通信协议。由MDC底层通信协议与应用层通信协议组成。

MDC通信支持RSA和3DES加密算法对数据进行加密， RSA加密采用2048位，3DES密钥采用24位。

MDC通信支持ZIP压缩算法对数据包压缩以较少通信数据量。

# 协议架构

**TCP/IP，** 通信协议底层，负责通信控制/路由/链路。

**MDC,** 协议包层，基于二进制流的通信协议,负责组包/分包，握手，能力交换，Key交换，压缩，加密，断点续传，数据包传输，

**App,** 应用通信层，负责处理应用请求，基于JSON + 二进制流的数据通信协议。

# MDC通信阶段



MDC通信协议分4个阶段：激活，hello，应用请求，bye个阶段。

**激活：**客户端发起，用于初始化客户端， 获取服务端的公钥和客户端的id 号。每客户端只需要调用一次。该阶段完成后，服务器端保存生成的clientid, 从客户端获取的uid, 客户端公钥；客户端保存从服务器获取clientid和服务器的公钥。

**Hello：**客户端发起，每次创建链接后，用于验证客户端的合法性和获取3DES加密key。

**应用请求：**客服端和服务器都可以发起，用于数据通信。

**Bye:** 客户端发起， 用于断开链接。

# MDC协议包

## 基本数据结构

* Byte, 1字节（8位）。表示{byte}。
* Short， 2字节，从左到右， 为short 的高到低。表示{short}。
* Int， 4字节，从左到右，为int 的高到低。表示{int}。
* Long， 8字节，从左到右，为long的高到低。表示{long}。
* Binary，二进制流。表示{binary}。通常都有{int}{binary}组成， {int}标识二进制流的长度。
* String, Int(长度)+二进制流（UTF-8）。表示{string}。

## MDC协议包结构

有4部分：包头，长度，命令，数据组成。

1）包头：1字节，从高位到底位：

* 协议标识：2bits, 固定值01
* 加密标识：1bit, 1加密，0未加密
* 压缩标识：1bit, 1压缩，0未压缩
* 保留：4bits

1. 长度：4字节，命令与数据包的长度

* {int}。

1. 命令与数据包：1字节
   1. 0x01:激活，JSON of {string}, 包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| capability | int | 是 | 设备能力，与包头位值对应 |
| uid | string | 是 | 设备唯一标识符号 |
| key | string | 可选 | 设备公钥，支持加密时用。 |

* 1. 0x80 + 0x01:激活回应，JSON of {string}, 包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| state | String | 是 | “ok”：成功  “fail”：错误 |
| clientid | string | 可选 |  |
| message | string | 可选 | 错误原因 |

* 1. 0x02: Hello 包， JSON of {string}, 包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| clientid | String | 是 | 激活时返回的clientid |
| uid | string | 是 | 如果支持加密， 则加密，加密算法：Base64(RSA(原始激活时的uid)) |
| capability | int | 否 | 客户端能力 |

* 1. 0x80 + 0x02: Hello 包， JSON of {string}, 包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| state | String | 是 | “ok”：成功  “fail”：失败 |
| code | String | 可选 | 握手成功，返回用于3DES的加密key, 为Base64(code) |
| message | string | 可选 | 错误原因 |

* 1. 0x03: Bye 包, 结束链接，清理资源， 无回应包。
  2. 0x04:断点续传二进制流包， JSON of {string} + {binary}, JSON包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| file | String | 是 | 为源端文件名 |
| length | Int | 必需 | 二进制包的长度 |
| total | Long | 必需 | 文件总字节数 |
| Position | long | 必需 | 本二进制包在文件中的位置 |

* 1. 0x80 + 0x04:断点续传回应包， JSON of {string} , JSON包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| State | string | 是 | “ok”:成功  “fail”：错误 |
| File | String | 是 | 文件名 |
| Message | String | 可选 | “done”：所有块接收完毕 |
| List | Json array | 可选 | 缺失块的起至位置 |
| list.s | Long |  | 缺失块的开始位置 |
| list.e | long |  | 缺失块的结束位置 |

* 1. 0x05:应用数据报，JSON of {string} +{int}+ {binary}, JSON包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| uri | String | 是 | 应用URI |
| param | JSON | 可选 | 应用参数 |
|  |  |  |  |

* 1. 0x80 + 0x05:应用数据报，JSON of {string} + {binary}, JSON包含元素：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 类型 | 必需 | 说明 |
| Uri | String | 是 | 应用URI |
| Contenttype | string | 是 | 内容编码 |
| error | int | 可选 | 错误编码参见HTTP协议 |
| Message | string | 可选 | 错误消息 |
| data | string | 可选 | 返回数据 |
| <其他> |  |  | 应用返回的结果 |
|  |  |  |  |

# 应用协议

1. 服务请求
   1. 服务：与Web服务结合， 使用同HTTP请求相同的URI, 当收到一个服务器请求包， 应用调用服务器的onMDC 方法，并传递param作为服务器的query参数。
   2. /hello: 新链接， 成功hello之后，调用/hello。
   3. /bye: 当接收到bye 命令或链接被断开后， 调用/bye
2. 应用包中的二进制
   1. 对于实时性要求高的应用(如视频或对话)，在每个应用包的JSON {string}后面都可以附带一段二进制流，二进制流以{int} + {binary} 的方式，{int}为后续二进制流的长度。