Push Protocol

1. 角色

以broker 作为server 端
app client 和 worker 皆为客户端

client 向 server 端发送的命令为纯文本内容方便利用 nc/telnet/tcpdump 等工具进行调度与查看。
连接建立后，client 必须发送一个4字节的"magic" 的标识通信协议的版本。
client 每隔5分钟 向服务端发送心掉，服务端回复同样的心跳信息。 client 发送心跳未收到回复时，将尝试重新连接 server.

2. TCP 协议

版本号约定

`[space][space][V][1]`

客户端心跳包

`H\n`

IDENTIFY

更新客户端元数据，协商连接的特性（如：心跳间隔、SSL 或 启用压缩等）

```
IDENTIFY\n
[ 4-byte size in bytes ][ N-byte JSON data ]
```

json data

`{"client_id": client_id, "heartbeat_interval": 300}` 其中心跳时间以秒为单位

SUB

client 订阅指定的channel

`SUB[space]<channel_id>\ n`

`<channel_id> - 订阅频道的ID(int64)`

Success Response:

`OK`

Error Response:

`E_INVALID_CLIENT`
`E_BAD_CHANNEL`

PUB

client 发送消息给指定的client, channel

```
PUB[space]<client_id> [space]<channel_id>[space]<message_id>\n
[ 4-byte size in bytes ][ N-byte message data ]
```

`<client_id> - 目标client的ID(int64)`
`<channel_id> - 目标channel的ID(int64)`
`<message_id> - message的ID(int64)`

Success Response:

`ACK 1 <message_id> <client_id>`

Error Response:

`ACK 0 <message_id> <client_id>`

CLS

client 关闭连接, 收到服务端成功返回后即可关闭自己的连接

`CLS\n`

Success Responses:

`CLOSE_WAIT`

Error Responses:

`E_INVALID`

数据格式

定义为数据帧的结构以支持不同的内容

```
[x][x][x][x][x][x][x][x][x][x][x][x]...
|  (int32) ||  (int32) || (binary)
|  4-byte  ||  4-byte  || N-byte
------------------------------------...
    size     frame type     data
```

client 会收到如下的数据帧类型:

```
    // when successful
    FrameTypeResponse int32 = 0
    // when an error occurred
    FrameTypeError int32 = 1
    // when it's a serialized message
    FrameTypeMessage int32 = 2
    // when ack a put message success/failure
    FrameTypeAck int32 = 3
```

message 的格式

```
[x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x][x]
[x][x][x][x][x]...
|       (int64)        ||    ||      (hex string encoded in ASCII)
|| (binary)
|       8-byte         ||    ||                 16-byte
|| N-byte
------------------------------------------------------------------------
--------------...
  nanosecond timestamp    ^^                   message ID
message body
                       (uint16)
                        2-byte
                       attempts
```

3.HTTP协议

3.1 client 注册设备

Router负责设备的注册并分配给设备一个可连接的Broker地址。

URL: /registration
Host: 10.10.79.134:4171 Method: POST
Parmas:

- device_type
- device_name(设备名称)
- serial_no（手机序列号）

`const ( ALLDevice = 0 Browser = 1 PC = 2 Android = 3 IOS = 4 WindowsPhone = 5 Other = 6 )`

返回

成功 `{broker:"10.10.79.134:8600", device_id:1030391111929}//iOS设备不需要返回broker`

错误

```
    NotFound      = -1
    OK            = 0
    ParamErr      = 1
    InternalErr   = 2
    MethodErr     = 3
    MsgEmpty      = 4
    MsgTooLong    = 5
    MsgErr        = 6
    ChannelIdErr  = 7
    DeviceTypeErr = 8
    SerialNoErr   = 9
```

示例
`curl -X POST /registration?serial_no=SOHUNO20140401XX&device_type=3&device_name=搜狐Android测试机`

3.2 业务逻辑发消息

业务线推送消息给client
URL: /put
Host:10.10.79.134:8710 Method: POST
Parmas:

- channel_id 推送频道的ID
- device_type 目标设备类型（0：Android，1：iOS）

Body: 推送消息的内容

成功 `{"code":0,"msg":"OK","data":null}`

错误

```
    NotFound      = -1
    OK            = 0
    ParamErr      = 1
    InternalErr   = 2
    MethodErr     = 3
    MsgEmpty      = 4
    MsgTooLong    = 5
    MsgErr        = 6
    ChannelIdErr  = 7
    DeviceTypeErr = 8
    NoSubErr      = 9
```

4.1 管理后台

```
http port = 4173
```

参考资料

[NSQ TCP Protocol Spec]