尚云互联 P2P 软件
技术白皮书

Yichen Wu

2017/02/22

尚云互联

深圳市尚云互联技术有限公司

版本信息

最新版本:

- API:
  - C++(Win32/Linux/Android/iOS/OSX)：2.2.4.0
  - C (Linux)：2.0.0.1
- P2P Server：1.7.0
- Relay Server:：2.0.0

尚云互联

修订历史

Version	Date	Author	Modification
1.0.0	2014/05/01	Charlie Chang	最初版本
2.0.0	2016/12/28	Yichen Wu	1.增加目录，删除内容大纲。 2.修改文档字体和段落布局。 3.转发服务器程序篇，安装架设章节,参数说明中，增加参数 RelayIP、修改展示图片。 4.平台测试工具篇修改范例的文字展示改为图片展示测试结果。
2.1.0	2017/02/22	Yichen Wu	1.更改平台测试工具篇，ConnectionTester 中的部分截图。 2.删除 P2P Server 篇--Optional Function 章节
Note: Do please update this table each time you change this document			

深圳市尚云互联技术有限公司保留文档内容的最后解释修改权利。

目录

尚云互联

DID 篇

一、DID 编码规则

- DID 格式为：XXXX-123456-ABCDE，其中：
  - XXXX：前置码(前缀码),可以用来区分不同产品,客户或销售区域等
    - 不同前置码的转发服务器(含设备转发)彼此不能共享，亦即每个前置码必须独立架设，
    - 前置码数量不宜过多(建议不要超过 40 个)，因为过多前置码数量会影响服务器的效能 前置码统一用大写,不得用数字,建议不要用英文字母“O”，前置码最长为 7 码,最少为 3 码
  - 123456：流水号，六位数字，000000~999999
  - ABCDE：检查码
    - 检查码是根据前置码及流水号通过私有算法而得来的
    - 服务器上会查核检查码是否正确
- 等效 DID 输入：以下的 DID 都等同，XXXX-123456-ABCDE，XXXX123456ABCDE，XXXX123456-ABCDE 或 XXXX-123456ABCDE

二、APILicense Code

- APILicense Code 是根据 DID 加上私有演算出来的
  - DID 的检查码是在 P2P Server 上验证的
    - 目的是确保 DID 是合法授权的
  - APILicense 是在 API 内验证的
    - 目的是保护 API 的合法使用
    - APILicense 只有在 Listen 的时候需要，

Listen(DID,....,APILicense)

- Client 连接设备时不需要 APILicense
- 如果 APILicense 验证错误就会返回-21 ERROR_PPPP_INVALID_APILICENSE（-21）

三、CRCKey

- CRC Key 是设计来保护 P2P 平台拥有者之权益的安全机制
  - 不知道正确的 CRCKey，即使是正确的 DID 也是无法登入 P2PServer
  - 避免正确 DID 的假设备来影响真设备
    - 真 DID：该 DID 确实是合法正确的(但可能透过非法的管道取得)
    - 假设备：不是真正的设备制造者(平台拥有者)所伪装出来的设备
    - 真设备：由原厂卖出并带有合法 DID 的设备
- 客户自行在其 P2P Server 上设定 CRCKey
  - 三台 P2P Server 上设置的必须是一样的
  - 长度不限，但建议是 6~8 的字母，太长则会影响效能
  - 尽量不要告诉任何人该 CRCKey，包括尚云互联以及设备的设计者
- CRCKey 是在 Listen()的时候输入，并置于最后一个参数与 APILicense 一起，按照 ”AAAAAA:CCCCCC”之格式，其中：AAAAAA 是 APILicense，CCCCCC 是 CRC Key，中间用“:”隔开

四、设备 Flash 内存里的参数区

- 一台采用尚云互联 P2P 软件平台的设备会需要以下与 P2P 有关的参数配置区：
  - DID：每个设备的唯一 ID

- APILicense：不同的 DID 有不同的 APILicense
- *CRCKey：CRCKey 是所有设备都一样的
- *InitString：服务器地址加密后的字符串,所有设备都一样
- *注：同一个平台里面的所有设备，其 CRCKey 跟 InitString 都一样,故 也可以在程序里面写死。
- 如果既有的参数区只有 DID，则可以采这样做法 XXXX-123456-ABCDE,AAAAA:CCCCCCC
- DID 与”APILicense:CRCKey”间用”,”分开,调用方式 Listen(“DID”，..... ,“APILicense:CRCKey”)

尚云互联

P2PServer 篇

一、安装架设

- P2P Server 最多可以有三台，建议一开始就是三台全都架好
- 关于 P2P Server 的规格要求:
  - OS: 32 or 64-bit Linux OS (Debian, Ubuntu, Redhat, Fedora..etc，为了避免不必要的麻烦,建议使用 Ubuntu,本手册之内容也是以 Ubuntu 为基础所撰写)
  - Hardware(CPU/Memory/Storage):一般云主机的最低规格即可，如果不够后续再扩充即可。
  - 带宽大小是关键: 1 百万的在线数量大约需要@100 Mbps 的带宽
  - 固定的公网 IP (尽量不要有异动 IP 的情况，如不能保证则请用域名
    - ■ 如果有防火墙,请务必取消 UDP 的阻挡
  - P2P Server 应该架在不同的机房甚至不同的地区，才具有备援效果
- 安装步骤
  - 准备好三台 P2P 服务器主机或云主机(建议云主机)
  - 将 P2P Server 程序放入目录位置(例如:/root/PPC)，(**libmysqlclient.so.18 也要放进相同目录下)
  - 修改配置文件: P2P Server 的配置有 **Server.conf**, **RMPassword.conf**, **BlackList.conf**，请参考“P2PServer - 设定配置档案”章节
  - 建立 Log 目录(mkdir /root/PPC/Log)
  - 将 P2P Server 程序设为开机后自动执行，其执行命令为“/root/PPC/PPCS_Server CRCKey &”
    - ■ 建议使用提供的 start_Server.sh (内容参考如下)
    - ■ 建议使用 root 权限

- 请注意 P2P Server 的执行权限
- CRCKey 为用户自行设置，建议 6~8 个字母或数字，区分大小写，请参考”P2PServer CRCKey” 章节

- start_Server.sh（红色字会因用户实际使用而不同）

```
#!/bin/bash
cd /root/PPC
s=1
while [ "$s" != "0" ]
do
    if [ $(ps -A | grep -c  'PPC_Server') == "0" ]; then
    /root/PPC/PPC_Server CRCKey &
    fi
sleep 5
done
```

- stop_Server.sh

```
killall start_Server.sh
killall PPC_Server
```

- 开机自动执行(根据实际情况会有变化,详细请看 P2P 部署文档)

```
# chmod 755 /root/PPC/*
#echo “/root/PPC/start_Server.sh&” >/etc/init.d/startPPC.sh
#chmod 755 /etc/init.d/startPPC.sh
#cd /etc/rc3.d #cd /etc/rc3.d
#ln -s ../init.d/startPPC.sh S81startPPC.sh
```

二、设定配置档案

- PPC 的 P2P Server 的配置档案有三:

  Server.conf, RMPassword.conf, BlackList.conf

  - 必须放置于 P2P Server 程序的目录,如: /root/PPC/
  - 皆为纯文本格式，请用 vim 修改
  - 如有修改必须重新执行 P2P Server
  - 基于安全考虑,请将/root/PPC 设为只有 root 能 access

- Server.conf: 用来设定支持的前缀码与流水号数量上限

- 档案内容为加密过之纯文本字符串
- 字符串内容请与尚云公司联系取得，修改将导致无效
- Server.conf 是必要的配置文件案，否则 P2PServer 是无法执行的

• RMPassword.conf:用来设定远程监看管理密码

- 档案内容由客户自行设定，建议至少为 8 码,英文大小写与数字之组合
- 如果不用远程监看管理功能,可以不要有 RMPassword.conf

• BlackList.conf: 用来取消某些特定 ID 号的 p2p 联机能力

- 某些 ID 号->根据前缀跟流水号来指定
- 如果没有需要禁用的 ID 号，可以不要有 BlackList.conf

• Server.conf (注意，您的内容应与以下的范例有所不同)

```
root@Debian:~/PPC#
root@Debian:~/PPC# cat Server.conf
CALCLCNJBBDILPPEFmcJGF1FdFGEsdfeDFsfDGLofGgOCDcGaFiIeKoAcMgFMFoNgI
root@Debian:~/PPC#
```

• RMPassword. (注意，您的内容应与以下的范例有所不同)

```
root@Debian:~/PPC#
root@Debian:~/PPC# cat RMPasssword.conf
A1b2c3D4
root@Debian:~/PPC#
```

• BlackList.conf. (注意，您的内容应与以下的范例有所不同)

```
root@Debian:~/PPC#
root@Debian:~/PPC# cat BlackList.conf
[GENERAL]
ListNo=2
List_0=AABB
List_1=AACC
[AABB]
No=3
SN_0=000099
SN_1=000098
SN_2=000097
[AACC]
No=1
SN_0=000123
```

三、CRCKey

- 关于 CRCKey:
  - CRC Key 是用来设计保护尚云互联 P2P 软件的用户，避免遭受假设备(但 DID 是对的)登入其 P2P Server 并造成真正的设备无法正常使用的情况(此一情况称之为假设备攻击行为)
  - 任何人在不知道正确的 CRCKey 的情况下是无法登入服务器的:
    - CRCKey 会直接影响设备的注册封包内容，而不是仅仅作为一个 CRC Checksum 来使用,故即 使是相同的 DID 只要 CRCKey 不同的话，其注册包内容也是完全不一样的。
- 设定 CRC Key:
  - 设定方式是在 P2P Server 的执行时，当作执行参数给定:
  - 例如:以下的执行命令，其 CRCKey 为“Ab09cD12”
    - # /root/PPC/PPC_Server Ab09cD12 &
  - CRCKey 是区分大小写的
  - CRCKey 长度不限但是太长的 key 会影响效能,建议是 6~8 码
  - CRCKey 在设备上的设置,请参考 DID 篇“DID－CRC Key”内容
- 保护 CRCKey:
  - 由于 CRCKey 是作为对假设备攻击行为的一道保护，而且一旦设备发行之后将无法修改,故应该绝对保密,不能泄漏!

四、Log

- P2PServer 会产生数个纯文本内容的 Log 档案,包括: Server.inf, DataFlow.log, DDHH.log:
  - Server.inf: 存放 P2PServer 的执行参数
    - 目录位置: /dev/shm

- 内容包括：服务器名称，版本号，Debug 层级，最大允许转发服务器数量(For 个别前置码)，CRCKey，支持的前置码及数量等.
- 为了避免信息外泄，请将此档设为只有 root 能 access

- DataFlow.log:每小时的数据流量纪录
  - 目录位置:/root/PPC (因用户实际使用而不同)
  - 每个小时的准点纪录前一个小时的值 In & Out number of Byte.
- DDHH.log:设备注册，客户联机请求，转发服务器登入...等封包的实时纪录
  - 目录位置:/root/PPC/Log(/root/PPC 因用户实际使用而不同,Log 目录名称则固定)
  - DDHH ->DD: 00~31 (每个月的日期)HH:00~23 (小时)，亦即每小时一份档案

- 除了以上的纯文本Log (之所以为纯文本，其目的就是为了方便用户查看)还有一个 Log:LoginIDs.dat，是设计来给 P2P Server 自身使用的 Log.
  - LoginIDs.dat:存放所有设备最后一次成功登入的时间
    - 目录位置:/root/PPC (因用户实际使用而不同)
    - 该档是自动产生的请勿自行任意修改,以免发生意外错误
    - 删除该档,虽然不会影响 P2PServer 之运行,但是有可能会丧失所有设备的最后一次登入时间的纪录信息
- DDHH.log 之内容说明:
  - A. Device Login log:
    - Error:
      - Login [XXXX] 'DID' from IP: Port

        XXXX = Invalid Prefix→Prefix is not supported by this server

        XXXX = Invalid DID→This DID is not valid, because, it's check sum is not correct!

XXXX = Blacked DID→This DID is in black list.
XXXX =Login Out of Date→DID License is Out of Date.

- Successfully:
  - Login 'DID' from IP: Port, NATType=#, APIVersion:#.#.#, LocalAddr=IP: Port

    NATType:0: Unknown, 1: IP-Restricted cone NAT, 2: Port-Restricted Cone NAT, 3: Symmetric NAT
    API Version: The API version, #.#.#
    Local Address: Local address of Login device

- B. Device relay function
  - Error:
    - SDev Login [XXXX] 'DID' from IP:Port

      XXXX = refer to A. Device Login log
  - Successfully:
    - SDev Login 'DID' from IP:Port

- C. P2P Request log
  - Error:
    - P2PReq [XXXX] 'DID' from IP: Port

      XXXX = refer to A. Device Login log
  - Successfully:
    - P2PReq 'DID' from IP: Port, LocalAddr= IP: Port

- D. Relay Server Login
  - Error:
    - RSLogin Login [XXXX] 'SID' from IP: Port

      XXXX = refer to A. Device Login log
      SID: The Relay Server's ID
  - Successfully:
    - RSLogin Login 'SID' from IP: Port, BandWidth=##, UserNumber=## “

      BandWidth is not yet implemented, so,just ignore it!!

User Number is not yet implemented, so, just ignore it!!

- E. Relay Request
  - Error:
    - RLYReq [XXXX] 'DID' from IP: Port

      XXXX = refer to A. Device Login log
  - Successfully:
    - RLYReq 'DID' from IP:Port, RLYAddr=IP: Port

      RLYAddr: the relay service provider's IP: Port
- F. Relay server list request log
  - Error:
    - ListReq1 [XXXX] 'DID' from IP: Port

      XXXX = refer to A. Device Login log
  - Successfully:
    - No Log when successful.

五、Management Tool

- 在 P2PServer 上有两种 Management tools
  - PPC_Management: 提供在本机上的管理
    - Usage:
    - 如果用户愿意，可以在 P2P Server 的主机上安装 WebServer,并用 CGI 的方式来执行 PPC_Management 程序，然后把执行后的产出，如./Log/DID_Dump.html 透过 Web 输出，如此便可以达到远程用 web 来查询的效果
  - RemoteCommander, RemoteMonitor: 远程管理
    - RemoteCommander
    - Usage

      Usage: ./RemoteCommander ServerHostName 1 Password
      →To get P2P Server running information.

Usage: ./RemoteCommander ServerHostName 2 Password
→To get statistics data of login devices.
Usage: ./RemoteCommander ServerHostName 3 Password Prefix SN
→To get login status of certain device of SN with prefix
Usage: ./RemoteCommander ServerHostName 4 Password
→Send EXIT Command to Server
Usage: ./RemoteCommander ServerHostName 5 Password
→Get Server.conf from Server
Usage: ./RemoteCommander ServerHostName 6 Password Server_Conf_FileName
→Update Server.conf to Server

- 其中的 Password 是在“P2P Server -设定配置档案”章节中所谈到的 RMPassword.conf 内所设置
- 当用户有多套 P2P Server 的时候，建议尽量透过 RemoteCommander 来控制并管理,以避免人为的错误!

- RemoteMonitor:
  - 是把 RremoteCommander 的控制功能抽掉的版本，这是为了让不具备管理者权限但是却有想远程来了解服务器情况的人来使用!
    - Usage:

Usage: ./RemoteMonitor ServerHostName 1 Password →To get P2P Server running information.
Usage: ./RemoteMonitor ServerHostName 2 Password →To get statistics data of login devices.
Usage: ./RemoteMonitor ServerHostName 3 Password Prefix SN
→To get login status of certain device of SN with prefix

转发服务器篇

一、转发服务器之运作说明

- 转发服务器(Relay Server)上线之后会主动向 P2PServer 报到，因此转发服务器不需要有固定的 Internet IP，且可以随时换到流量相对便宜的云主机上来执行
- 转发服务器使用 UDP 端口来提供转发服务，所以转发服务器主机上的防火墙必须关闭所有 UDP 端口的双向阻挡,否则无法提供服务(尤其特别注意运营商外部的防火墙)
- 转发服务器其实只是一个 Linux 上的可执行程序
  - 一台主机上可以跑多个转发服务程序，等同有多台转发服务器之效果(当然总流量还是受限制)
  - 每一个转发服务程序只能支持一款 DID 前置码，支持不同 DID 前置的转发不能互相支持，例如:AABB 前置的转发服务器，不会为 AACC 前置的设备提供转发服务，至于要支持哪款特定的前置码,是由转发服务器程序的执行参数所给定，详见“转发服务器-转发服务器 ID”章节
  - 转发服务器可以随需要添加/删除/移动,服务器数量并没有理论上的上限，而且添加/删除/移动都是立即生效(1 分钟之内)
    - 添加:在云主机上执行新的 Relay Server 程序
    - 删除：在云主机上终止执行 Relay Server 程序
    - 移动:在旧的云主机上终止，并在新的云主机上执行起来(同样的转发服务器 ID)

二、安装架设

- 安装步骤
  - 准备好 32 or 64-bit x86 Linux 实体主机或云主机，为了避免不必要的麻烦,建议使用 Ubuntu,本手册之内容也是以 Ubuntu 为基础所撰写)

- 将 PPCRelay Server 程序放入目录位置(例如:/root/PPC)
- 建立 Log 目录:

  #mkdir /root/PPC/RSLog
- 执行 Relay 程序
  - 建议使用提供的 start_Relay.sh
  - 建议使用 root 权限
  - 请注意 Relay 程序的权限

```
root@ubuntu:PPC# ./PPCS_Relay
Usage: ./PPCS_Relay SID SID Port BandWidth MaxUser P2PServer1 P2PServer2 P2PServer3 [DelayTime_ms] RelayIP
        SID:The SID of this Relay Server
        Port: The Relay Service Port
        BandWidth: Bandwidth of this relay server, in unit of Kbps, if unknow type 0
        MaxUser: Max number of relay service user, if unlimit type 0
        P2PServer1: Domain name of 1st P2P Server 0
        P2PServer2: Domain name of 2nd P2P Server 0
        P2PServer3: Domain name of 3rd P2P Server 0
        DelayTime_ms: The response delay time in ms. Delay time is to fine ture server'response
                The smaller the quicker server response a relay request,thus, increase bandwidth usage
                Default value:0(if not specified),suggest value : 200, max value: 1000
        Example: ./PPCS_Relay RLY-0000-XXXXX 12300 10000 50 127.0.0.1 127.0.0.1 127.0.0.1 200 127.0.0.1
```

- 程序参数说明
  - Usage: PPCS_Relay SID Port BandWidth MaxUser P2PServer1 P2PServer2 P2PServer3 [DelayTime_ms] [RelayIP]
    - SID: The SID of this Relay Server(详见”转发服务器-转发服务器 ID” 章)
    - Port: The Relay Service Port
    - BandWidth: Bandwidth of this relay server, in unit of Kbps, if unknow type 0(目前只是预留参数，对于程序功能没有 实际的效用)
    - MaxUser: Max number of relay service user, if unlimit type 0(用来限制转发服务数量的最大值)
    - P2PServer1: IP or Domain name of 1st P2P Server
    - P2PServer2: IP or Domain name of 2nd P2P Server
    - P2PServer3: IP or Domain name of 3rd P2P Server
    - DelayTime_ms: The response delay time in ms. Delay

time is to fine ture server' response. The smaller the quicker server response a relay request, thus, increase bandwidth usage. Default value:0(if not specified), suggest value : 200, max value: 1000

- RelayIP: Local IP (当该参数设定后，P2P 服务器获取转发服务器的 IP 时会先获取这个参数，因此当 P2P 服务器于转发服务器在同一台云主机或在同一机房时是必须设置，两者不在同一台云主机或不在同一机房时建议留空)

- Example: ./PPCS_Relay RLY-0000-XXXXX 12300 10000 50 s1.p2p.com s2.p2p.com s3.p2p.com 200 140.112.123.45
- MaxUser 是设计来限制转发服务的最大数量:
  - 如果设为 100,则表示最多只能提供 100 个同时的转发服务
  - 这个参数可以用来限制最大的带宽使用量，尤其在某些共享但有允许上限带宽的云主机上特别有用(如果不限经 常会因为超用带宽被停用云主机)
- Delay_Time 是用来调整转发服务请求的响应时间，可以用来让设备转发优先，或者平衡多个服务器间的负载平衡
- start_Relay.sh（红色字会因用户实际使用而不同）

```
#!/bin/bash
cd /root/PPC
s=1
while [ "$s" != "0" ]
do
    if [ $(ps -A | grep -c 'PPC_Relay') == "0" ]; then
    /root/PPC/PPC_Relay AABB-000000-ABCDE 10001 5000 500
P2PServerIP1 P2PServerIP2 P2PServerIP3 200 RelayIP &
    /root/PPC/PPC_Relay AACC-000000-BCDEA 11000 5000 500
P2PServerIP1 P2PServerIP2 P2PServerIP3 200 RelayIP &
    /root/PPC/PPC_Relay AADD-000000-CDEAB 12000 5000 500
P2PServerIP1 P2PServerIP2 P2PServerIP3 200 RelayIP &
    fi
    sleep 5
done
```

- stop_Relay.sh

```
killall start_Relay.sh
killall PPC_Relay
```

三、转发服务器 ID

- 每个转发服务器(Relay Server)都需要一个唯一的 SID(Relay Server ID)来区别彼此，并且不仅仅如此而已，SID 还用来：
  - 指定要服务的 DID 前置码 = SID 的前置码
  - 验证该服务器是否合法，SID 也是类似 DID 有检查码，SID 的结构：XXXX-123456-ABCDE
    - XXXX ：前置码
    - 123456：流水号
    - ABCDE:检查码
  - 为啥转发服务器 ID 也需要有检查码?
    - 如果转发服务器不带检查码的话，则可能会遭受到 Fake Relay Server 的攻击
    - 转发服务器的 SID，请向尚云公司申请。

四、Log

- Relay Server 会产生数个纯文本内容的 Log 档案，包括：Relay.inf，DataFlow.log，DDHH.log:
  - 为了区别不同的 SID,Relay.inf，DataFlow.log 后面会加上 SID:
    “Relay.inf_XXXX-123456-ABCDE”，“DataFlow.log_XXXX-123456-ABCDE”
- Relay.inf：存放 Relay Server 执行时的设定内容
  - 目录位置:/dev/shm
  - 内容包括：版本号，Debug 层级，执行参数.
  - 为了避免信息外泄，请将此档设为只有 root 能读取

- DataFlow.log:每小时的数据流量纪录
  - 目录位置:/root/PPC (因用户实际使用而不同)
  - 每个小时的准点纪录前个小时的值 In & Out Byte.
- DDHH.log:转发服务的使用纪录，错误纪录等等
  - DDHH.log:
    - 目录位置:/root/PPC/RSLog (/root/PPC 因用户实际使用而不同，RSLog 目录名称则固定)
    - DDHH->DD: 00~31 (每个月的日期)HH:00~23 (小时)，亦即每小时一个档案
- Log 纪录内容都是可以直接判读的

五、设备转发功能

- 如果网络条件允许，并且设备上有开启设备转发功能(APIfunction-> Share_Bandwidth(1)) 则该设备会向 P2P 服务器注册成为一个可提供转发服务之设备(Super Device)
- 设备转发功能说明:
  - 每一个 Super Device 只能为一个客户端提供转发服务
  - Super Device 随时可以中断转发服务(Share_Bandwidth(0))
  - 设备转发服务的控制权是在设备上的固件程序,而非 P2P 的 API(API 里面只是提供该功能，要不要启用,何时关闭,都是由设备端的程序来调用决定)
  - 建议的使用方式:在设备没有客户端进来联机的时候可以打开(Share_Bandwidth(1)),一但有客户 端联机成功则关闭(Share_Bandwidth(0))
- 设备转发功能在某些地区，必须经过使用者之同意，否则可能有违法的风险
- 设备转发功能会不会有隐私泄漏的问题?
  - 理论上不会，因为尚云互联 P2P API 不提供侧录之功能，故用户无法透过 API 来取得设备转发之封包内容 ,也没有任何 log

纪录

- 如果产品本身的协议已经有具备加密，则更不可能因为设备转发而隐私泄漏

尚云互联

平台测试工具程序篇

一、ConnectionTester

- Linux-based stand-alone command-line Program
  - Linux based: Require x86 32 bit Linux OS to run
  - stand alone: 可独立执行不需要 P2P API 库(已经包在里面)
  - Command line: 命令行模式操作
- 用途:
  - 用来确定某个 DID 设备是否能被联机
    - 外网联机 or 内网联机
  - 在没有设备的情况下，也可以用来测试 DID 是否为合法 DID.
  - 测试转发服务器是否正常
  - 长时间测试设备的联机稳定度
- Usage:

```
dugu@ubuntu:/mnt/hgfs/Workstation/P2P文件/Tester$ ./ConnectionTester
ConnectionTester Verion: 2.4.0 Build 16/11/28 17:17
API Version: 2.2.4.0
Usage: ./ConnectionTester Mode DID InitString [Repeat]
        Mode: 0, No local LAN search, P2P then Relay for remote
              1, Local LAN search, P2P then Relay for remote
              2, No local LAN search, force Relay for remote
              3, Local LAN search, force Relay for remote
              4, Do Network Detect Only
              6, No local LAN search, force Server Relay for remote
        Repeat: Repeat times.(eg: 100),  If not speaified, connect once.
```

- 范例:
  - 对某个 DID 做外网联机测试

```
dugu@ubuntu:Tester$ ./ConnectionTester 0 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=61.141.152.217:62302, Mode=P2P, Time=0.662 (Se
c)
```

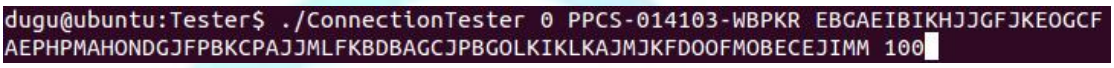
  - 在内网内对某个 DID 做联机测试

```
dugu@ubuntu:Tester$ ./ConnectionTester 1 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=192.168.30.128:25590, Mode=P2P, Time=1.518 (Se
c)
```

  - 即使 DID 设备并不存在，也可以用 ConnectionTester 来测试

DID 账号是否合法

- 长时间测试设备的联机稳定



- 用 ConnectionTester 来测试转发服务器是否正常

二、ListenTester

- Linux-based stand-alone command-line Program
  - Linux based: Require x86 32 bit Linux OS to run
    - stand alone:可独立执行不需要 P2P API 库(已经包在里面)
    - Command line: 命令行模式操作
    - 用途:
      - 用来侦测网络环境
      - 用来测试 DID 是否能成功登入到 P2P 服务器?

■CRCKey

◆ 搭配 ConnectionTester 来检测 APILicense 是否正确

- Usage:

```
dugu@ubuntu:Tester$ ./ListenTester_PPCS
ListenerTester Verion: 2.0.0 Build 16/12/04 12:22
API Version: 2.2.4.0
Usage: ./ListenTester_PPCS DID APILicense InitString
```

■ 备注:ListenTester 会等待 60 秒,如果没有 Client 连上则显示 timeout

- 范例:

- 用来侦测网络环境

```
dugu@ubuntu:Tester$ ./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView  EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
------------- NetInfo: ------------------
Internet Reachable     : YES
P2P Server IP resolved : YES
P2P Server Hello Ack   : YES
Local NAT Type         : Port-Restricted Cone
My Wan IP : [redacted]
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
```

- 用来测试 DID 是否能成功登入到 P2P 服务器

```
dugu@ubuntu:Tester$ ./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView  EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
------------- NetInfo: ------------------
Internet Reachable     : YES
P2P Server IP resolved : YES
P2P Server Hello Ack   : YES
Local NAT Type         : Port-Restricted Cone
My Wan IP : [redacted]
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
Got Server Response...
Listen failed (-3): Listen time out, No client connect me !!
dugu@ubuntu:Tester$ ./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyVie EBGAEIB
IKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
------------- NetInfo: ------------------
Internet Reachable     : YES
P2P Server IP resolved : YES
P2P Server Hello Ack   : YES
Local NAT Type         : Port-Restricted Cone
My Wan IP : [redacted]
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyVie')...
No Server Respnse!!!
```

登录成功

没登录成功，CRCKey错误?

- 搭配 ConnectionTester，来检测 APILicense 是否正确

```
dugu@ubuntu:Tester$ ./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView  EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
-------------- NetInfo: -------------------
Internet Reachable     : YES
P2P Server IP resolved : YES
P2P Server Hello Ack   : YES
Local NAT Type         : Port-Restricted Cone
My Wan IP : 61.141.152.217
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
Got Server Response...
DID : PPCS-014103-WBPKR
Remote Addr : 192.168.30.128:19383
Connection mode: P2P
```

```
dugu@ubuntu:Tester$ ./ConnectionTester 1 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=192.168.30.128:15641, Mode=P2P, Time=0.214 (Se
c)
```

连接成功则表示 APIlicense 正确。

Watch Dog 篇

一、工作原理

- Watch Dog 目的是用来监看服务器是否正常运作，包括 P2P 以及转发服务器
  - 每隔一段时间(约 15 sec)，WatchDog 会发送一个封包给的指定的服务器
  - 服务器收到之后会响应一个 ACK 封包
  - 如果连续五分钟都没有收到 ACK，则判定该服务器有异常 (Server Down)
- WatchDog 会发信件通知,所有服务器 Up / Down 事件
- WatchDog 可以架设一台以上，以防 WatchDog 本身出问题
- WatchDog 主机的规格需求
  - 32 Bit x86 Linux server with Stable Internet connection
  - Web Server
  - No UDP firewall

二、安装架设

- Step 1:准备好云主机(以下上的安装架设步骤皆为在 amazonEC2, t1.micro，AMI Linux 32bit 上，建议比照办理)，安装好 Web Server，并设定 80 port 打开
  - yum install httpd
- Step 2: 将 WatchDog 程序放到/root/WatchDog/下
- Step 3: 修改 WatchDog 配置档案的内容: MailList.cfg 及 WDT.cfg
  - MailList.cfg: 设定 Server Up/Down 邮件通知收件信箱
  - WDT.cfg: 设定要 Watch 的 Server IP, Port
- Step 4: 执行 WatchDog 程序，并设为开机自动执行

- 请使用 root 权限
- WatchDog 会产生一个文本文件 /var/www/cgi-bin/Wdt_Status.log, 如果 /var/www/cgi-bin/ 目录不存在,请手动 mkdir /var/www/cgi-bin/

- Step 5: 修改并设定监看服务器状态用的 ServerMonitor.cgi
  - 请注意 ServerMonitor.cgi 的执行权限
  - 请修改 ServerMonitor.cgi 的内容,以能够将 /var/www/cgi-bin/Wdt_Status.log 给输出到 browser 上
  - 请将/var/www/cgi-bin/Wdt_Status.log 设为所有人可以读
    - Chmod 644 /var/www/cgi-bin/Wdt_Status.log
- config file:
- MailList.cfg 内容

```
/root/WatchDog$ cat MailList.cfg
abcd@163.com
abcd@gmail.com
/root/WatchDog$
```

- WDT.cfg 内容 （Format: IP:Port,Descriptions）

```
/root/WatchDog$ cat WDT.cfg
12.125.213.100:32100,PPC P2P Server 1
12.125.212.208:32100,PPC P2P Server 2
12.125.211.207:32100,PPC P2P Server 3
12.125.211.217:10001,PPC Relay Server 1 [PPC]
12.125.211.218:10001,PPC Relay Server 2 [PPC]
/root/WatchDog$
```

- ServerMonitor.cgi:
  - ServerMonitor.cgi 是为了方便可以随时查看从 WatchDog 上监测到的服务器状态
    - http://your.watchdog.ip or domain/cgi-bin/ServerMonitor.cgi
  - ServerMonitor.cgi 的内容:

```
#!/bin/bash
```

```
echo "Content-type: text/html"
echo ""
echo '<html>'
Echo '<head><meta http-equiv="content-type"
content="text/html; charset=UTF-8">'
echo '<meta http-equiv="refresh" content="60;
url=cgi-bin/ServerMonitor.cgi">'
echo '<meta http-equiv="refresh" content="60;
url=cgi-bin/ServerMonitor.cgi">'
echo '<title>PPC Server Monitor</title></head>'
echo '<body><h1>Server Running Status</h1><hr />'
echo '<pre>'
cat Wdt_Status.log
echo '</pre></body></html>'
exit 0
```

- 请注意 ServerMonitor.cgi 的执行权限，目录位置，以及 /var/www/cgi-bin/Wdt_Status.log 的 读取权限

Remote Management 篇

一、工作原理

- Remote Management 是指从远程的主机去控制或者察看 P2P Server 运作的功能:
  - Remote Management 程序:RemoteCommander & RemoteMonitor
    - RemoteCommander:兼具控制与察看的功能
    - RemoteMonitor:只能察看
- 要执行 Remote Management 需要密码
  - 请参考章节“P2P Server -设定配置档案” 里面的 RMPassword.conf
- Usage:
  - RemoteCommander

    Usage: ./RemoteCommander ServerHostName 1 Password --> To get P2P Server running information.
    Usage: ./RemoteCommander ServerHostName 2 Password --> To get statistics data of login devices.
    Usage: ./RemoteCommander ServerHostName 3 Password Prefix SN --> To get login status of certain device of SN with prefix
    Usage: ./RemoteCommander ServerHostName 4 Password --> Send EXIT Command to Server
    Usage: ./RemoteCommander ServerHostName 5 Password --> Get Server.conf from Server
    Usage: ./RemoteCommander ServerHostName 6 Password Server_Conf_FileName--> Update Server.conf to Server

  - RemoteMonitor

    Usage: ./RemoteMonitor ServerHostName 1 Password --> To get P2P Server running information.
    Usage: ./RemoteMonitor ServerHostName 2 Password --> To get statistics data of login devices.
    Usage: ./RemoteMonitor ServerHostName 3 Password Prefix SN --> To get login status of certain device of SN with prefix c

二、安装架设

- 需求:要执行 Remote Management 之主机必须有以下的规格
  - OS:x86 Linux , 32-bit, Internet accessable
  - 允许 UDP 封包往外收送
- 执行 RemoteCommander & RemoteMonitor
  - RemoteCommander, RemoteMonitor 只是标准的 Linuxconsole mode
  - 在 shell 下直接执行

三、安全性

- Remote Management 功能由于牵涉到服务器的安全, 必须特别注意
  - 请勿泄漏 RMPassword.conf 里面的密码
  - 如有疑虑应立即修改 RMPassword.conf 里面的密码, 密码修改后需重启 P2P 服务器程序方能生效
  - RemoteCommander 具备控制能力,应只能给最高权限管理者使用

尚云互联

API 篇

一、初始化字符串

- 初始化字符串或称服务器地址字符串
  - 是内容经过加密之英文字母字符串
    - 由尚云公司提供
    - 修改内容会造成 API 无法正常使用
  - 用来指定 P2P 服务器的域名或者 IP(InitString)
    - PPCS_Initialize(“XXXXXXXXX”)
    - PPCS_ConnectByServer(....., “XXXXXXXXX”)
    - PPCS_NetworkDetectByServer(..... , “XXXXXXXX”)

二、Listen & Connect

- 功能相当于 TCP/IP Socket 的 Listen & Connect
  - Device 调用 Listen()来等待 Client 的 Connect()
  - Listen()跟 Connect() 都需要 DID
- Listen & Connect 会阻塞程序的执行,直到联机完成,超时或者异常的错误发生
  - 联机完成会返回一个>=0 的 Session 的 ID
    - PPCS_Check()可以察看 Session 的详细信息
    - Session 上有 8 个 Channel 可以用来 Read/Write 数据, Channel 间彼此独立的
    - PPCS_Close()可以将一个 Session 关闭
    - PPCS_ForceClose()强制关闭一个 Session,不管资料是否传完
  - 如果超时或失败则会返回负的值

三、Write 与流量控制

- PPCS_Write()是立即返回的
  - 因为 API 内部有 Thread 负责传送
- 透过 PPCS_Write()写入的数据是保证送达的
  - 除非 Session 断线(Timeout, Remote or Local Close)
- PPCS_Check_Buffer()可以得知 Write 入的数据还有多少没送出去
- 流量控制做法:
  - 在 PPCS_Write()之前先 Check Write Buffer 数量
  - 如果 Buffer 数量大于某个值，则暂停写入并 Sleep 等待
  - 在 LiveVideo 的应用下，可以考虑将 Videoframe 扔掉
    - 因为这个时候网络传输速率小于 Video 产生的速率

四、Read 与 Time Out

- PPCS_Read()给定 Timeout 时间，会 block 执行到时间结束或预期要读到的数据长度已经满足
  - 返回 Timeout 有可能已经有部分数据已经读到，必须判断返回的 DataSize 才能得知切确读到的资料长度

五、库与版本发布

- PPCS_GetAPIVersion() 可以获取 API 的版本号
- PPCS_GetAPIVersion() Returns 0x01020304 -> Version 1.2.3.4
- 从 1.0.0.0 以后,版本号的最后一码是用来分辨 API 是 C 还是 C++
  - X.X.X.0 -> C++ 版本
  - X.X.X.1 -> C 版本

六、兼容性

- 除非特别声明，尚云互联 P2P 的 API 都是往前兼容，也就是说：
  - 新的 Client 可以支持旧的 Device
  - 旧的 Client 也可以支持新的 Device
- 举例来说：
  - 使用 1.5.1 版的 App 可以支持 0.2.8 的设备
  - 使用 0.2.8 版的 App 可以支持 1.5.1 的设备
    - 也就是说 APP 可不用更新，仍然可以 Support 新的版本的设备

尚云互联