

中国电信集团公司企业标准

Q/CT X-2017

家庭终端与智能家庭网关自动连接的接口 技术要求

Technical Specification of Interface for Automatic Connection between Home Terminal and Smart Home Gateway

2016-××发布 2016-××实施

中国电信集团公司 发布

家庭终端与智能家庭网关自动连接的接口技术要求	
1 范围	2
2 规范性引用文件	2
3 缩略语 2	2
4 总体定义	3
4.1 家庭网络中家庭终端的类型及定义	
4.2 接口定义	4
5 功能要求	4
5.1 智能家庭网关设备功能需求	
5.2 组网终端设备功能需求	
5.3 应用终端设备功能需求	
5. 4 手机 APP 功能需求	
6 交互流程定义	
6.1 组网终端接入智能家庭网关及平台的实现流程	
6.2 应用终端快速接入智能家庭网关实现流程 10	
7 接口定义	1
7.1 Tn接口——智能家庭网关与组网终端的同步接口12	2
7.2 Ta接口——智能家庭网关与应用终端的实现接口24	4
附 录 A 组网终端状态变迁图27	7

家庭终端与智能家庭网关自动连接的接口技术要求

1 范围

本标准适用于中国电信家庭网络中使用的智能家庭网关、组网终端(无线AP、支持AP功能的电力猫(PLC),无线中继(Repeater)等)和应用终端(WiFi无线方式上连的智能家居设备等)。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅所注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

Q/CT 2603-2015 中国电信智能家庭网关技术要求

Q/CT 2604-2015 中国电信智能家庭网关与智能网关及应用管理平台接口技术要求

3 缩略语

下列缩略语适用于本文件。

缩略语 英文全称 中文全称

AP Access Point 无线接入点

APP Application 手机等移动智能终端的软件客户端

AES Advanced Encryption Standard 高级加密标准,详见注1

Big

Big endian 大端格式,将高序字节存储在起始地址(高位编址)

endian

是一种轻量级的数据交换格式,是对 Json 格式的字

符串进行构建和解析的一个C语言函数库。本协议

c JSON c JavaScript Object Notation 采用的 cJSON 为 sourceforge 上的开源软件,更新

时间为 2013-08-14 的版本。可以通过

<u>http://sourceforge.net/projects/cjson</u>获得。

DH Diffie-Hellman 迪菲-赫尔曼秘钥交换算法,详见注2

Dynamic Host Configuration

DHCP 动态主机配置协议

Protocol

EoC Ethernet Over Cable 通过同轴电缆传输宽带数据的技术

互联网协议 Internet Protocol Television 交互式网络电视 **TPTV** LED LightEmittingDiode 发光二极管 MAC Media Access Control 物理地址 Personal Identification PIN 个人识别密码 Number WiFi Repeater Repeater 无线中继器 Simple Service Discovery SSDP 简单服务发现协议 Protocol UI User Interface 用户界面 UPnP Universal Plug and Play 统一即插即用协议 VLAN VirtualLocalAreaNetwork 虚拟局域网 WAN Wide Area Network 广域网 WiFi Wireless-FIdelity 无线保真 WLAN Wireless LAN 无线局域网

WiFi Protected Setup

Internet Protocol

注1: AES: 本协议中采用AES的CBC模式进行加解密, AES-CBC算法采用开源软件openss1-1.0.1的支持, 密钥长度为 128bit, 16字节, 当密钥不足16字节长度时, 将密钥尾端按全0补足; 例如: 112bit, 14字节的原密钥 0102030405060708090a0b0c0d0e (数 组 十 六 进 制 格 式 {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e})补齐为16字节AES-CBC密钥 0102030405060708090a0b0c0d0e0000 (数 组 十 六 进 制 格 式 : {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x00, 0x00}); 当密钥长度 超过128bit (16字节), 仅截取前128bit (16字节)内容作为加解密的KEY。

WiFi保护设置

注2:一种确保共享KEY安全穿越不安全网络的方法,需要安全通信的双方可以用这个方法确定对称密钥。然后可以 用这个密钥进行加密和解密。本协议中使用128bit的公钥。

4 总体定义

WPS

ΙP

4.1 家庭网络中家庭终端的类型及定义

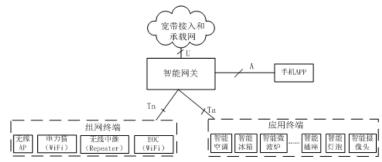


图1 家庭网络系统组网图

如图1所示,智能家庭网关是家庭网络连接外部网络的接口单元,上连宽带接入和承载网,下连家庭终端。本标准中的家庭终端包括:组网终端和应用终端。组网终端除无线中继(Repeater)外均采用有线方式接入智能家庭网关,应用终端具备WiFi接入能力,通过WiFi无线方式接入智能家庭网关。

家庭终端: 是指在家庭环境内需要接入互联网的终端设备,本标准主要是指组网终端和应用终端两类;

组网终端:是指在家庭范围内延伸智能家庭网关网络覆盖的终端设备,主要包括:无线AP、电力猫、无线中继以及具备WiFi能力的EOC终端设备等;

应用终端: 是指在家庭范围内通过WiFi接入的智能家居终端设备。

4.2 接口定义

a) U接口

U接口是智能家庭网关与宽带接入网的接口,关于U接口的详细定义和要求,参见《中国电信智能家庭网关技术要求》(Q/CT 2603-2015)。

b) A 接口

A接口是手机APP和智能家庭网关之间的本地逻辑接口,可以实现对智能家庭网关及家庭终端的配置及管理功能,关于A接口的详细定义和要求,参见《中国电信智能家庭网关技术要求》(Q/CT 2603-2015)。

c) Tn接口

Tn接口是智能家庭网关与组网终端之间的无线配置信息同步实现接口,用于传输注册认证和无线配置参数下发等控制消息,可采用以太网有线/WiFi无线承载方式。

d) Ta接口

Ta接口是通过手机APP实现的智能家庭网关及组网终端与应用终端之间的WiFi无线快速接入(WiFi无线快速接入,以下简称"快连")及应用终端设备的发现接口,用于传输无线快速连接的协商及智能应用终端的设备发现等控制消息。

5 功能要求

5.1 智能家庭网关设备功能需求

智能家庭网关为了支持Tn接口需满足如下功能要求:

- a) DH密钥交换功能;
- b) 智能家庭网关验证组网终端合法功能;

4

- c) 智能家庭网关具备向手机 APP 请求确认组网终端的无线信息同步功能;
- d) 智能家庭网关保存已确认的组网终端信息。组网终端下线后,发起对同一智能家庭网关的注册 请求,智能家庭网关已确认列表中查找确认过的组网终端,如果找到,将直接发送配置信息给 组网终端,而不需要再次请求用户确认:
- e) 智能家庭网关具备将自身无线配置信息发送给组网终端的功能,该功能可配;
- f) 智能家庭网关查询配置信息改变功能;
- g) 智能家庭网关具备向管理平台报告下挂组网终端、应用终端相关设备信息和状态参数的功能。 智能家庭网关为了支持Ta接口需满足如下功能要求:
- a) 智能家庭网关内置支持 WPS 功能;
- b) 支持通过 APP 方式启动智能家庭网关 WPS 操作;
- c) 支持通过 APP 查询 WPS 自动配置状态(包括已进入 WPS 配置状态、WPS 启动失败、有设备通过 WPS 过程接入网络、WPS 关闭等);
- d) 支持通过 APP 中止 WPS 操作;
- e) 智能家庭网关应关闭 WPS PIN 验证方式;
- f) 智能家庭网关应支持在 WiFi 网络中转发来自手机 APP 的 UPnP SSDP 多播数据报文功能;
- g) WPS 协商过程中,智能家庭网关 WiFi 网络中已经连接的设备不应断开。

5.2 组网终端设备功能需求

- a) DH 密钥交换功能;
- b) 组网终端设备信息提交、设备注册功能;
- c) 组网终端设备配置功能,且对配置/查询操作响应时间不高于3秒;
- d) 组网终端设备链路保持发起功能;
- e) 组网终端设备路由/桥接模式的切换功能(Repeater 除外);
- f) 组网终端设备与智能家庭网关同步后自动切换到桥接模式的功能;
- g) 在桥接工作模式下,组网终端设备的配置管理 IP 地址仍有效;
- h) 组网终端设备在桥接模式下应能支持对下挂应用终端的 MAC 地址、连接类型等信息进行透传;
- i) 组网终端设备在桥接模式下应支持 VLAN 透传(承载 IPTV 业务需求等);
- j) 本地 WEB 管理页面支持"启用与智能家庭网关的无线配置同步"管理选项,用于满足用户选择是否启用与智能家庭网关的无线配置进行同步,缺省自动启用;
- k) 应支持 WPS, 能够响应智能网关下发的 WPS 开关信息, 启动/关闭组网终端设备自身的 WPS 连接功能;
- 1) 应支持通过集成云客户端实现和电信 AP 集约管理平台的通信,并支持对云客户端的单独升级。

5.3 应用终端设备功能需求

- m) 支持内置 WPS 功能;
- n) WPS 功能的触发可采用多种方式,如 WPS 物理按键、恢复出厂后重新加电启动等;
- o) 能响应来自手机 APP 的快连报文,并主动完成设备功能上报(类型、名称、厂家、序列号、IP 地址、其他设备辅助信息等)。若已与手机 APP 快连成功,则不再响应来自手机 APP 的快连报文,除非再次触发 WPS。

5.4 手机 APP 功能需求

- a) 与智能家庭网关建立指令连接通道,传递交互指令;
- b) 发送指令,触发智能家庭网关 WPS 功能,并接收或查询执行状态;
- c) 发送指令,中止智能家庭网关 WPS 过程;
- d) 支持周期式发送多播指令(SSDP 报文,多播地址及端口号为: 239.255.255.250:1900),包含手机在 WiFi 网络中的 IP 地址等信息,并接收新设备上报的数据;
- e) 及时在 UI 界面上更新操作执行的信息反馈;
- f) 手机 APP 具备 Ta 接口的快连操作引导(正在进行什么操作、提示用户接下来如何操作)功能;
- g) 需支持基于 SSDP 的设备发现流程和数据接口(详见 6.2、7.2)。

6 交互流程定义

本交互流程定义家庭终端与智能家庭网关的交互过程,包括:组网终端与智能家庭网关的通信流程、应用终端无线自动接入智能家庭网关的实现流程。

6.1 组网终端接入智能家庭网关及平台的实现流程

根据组网终端接入场景,分为如下两种情况:

- a) 组网终端接在工作在路由模式的智能家庭网关下,需要同时和智能家庭网关以及 AP 管理平台建立通信连接(如图 2 所示)。
- b) 组网终端接在非智能家庭网关下或者工作在桥接模式的智能家庭网关下,需要和 AP 管理平台建立通信连接。

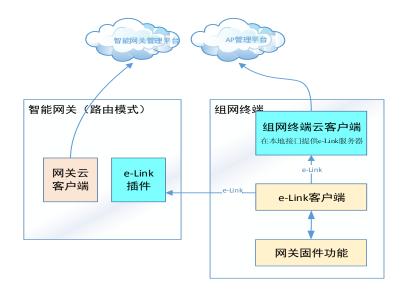


图2 组网终端接入场景

6.1.1 组网终端接入智能家庭网关实现流程

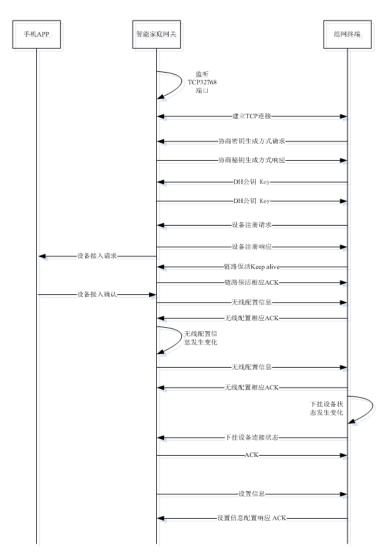


图3 组网终端与智能家庭网关无线参数自同步流程图

流程说明:

- a) 智能家庭网关上电, 打开 TCP 32768 端口, 等待外部连接;
- b) 组网终端上电,通过 WAN 口以 DHCP 方式获取到 IP 后,连接智能家庭网关的 TCP32768 端口;
- c) 组网终端发向智能家庭网关发起协商密钥生成方式请求,将组网终端侧支持的密钥生成方式 (DH) 发送给智能家庭网关;

- d) 智能家庭网关在收到的组网终端的密钥生成方式中,选择智能家庭网关支持的最优密钥生成方式(DH),并回应给组网终端,完成双方密钥生成方式协商。
- e) 组网终端侧采用 DH 算法,生成公钥和私钥,发送组网终端的公钥与组网终端的设备信息给智能家庭网关;
- f) 智能家庭网关通过 TCP 32768 端口收到对方的 DH 公钥后,采用 DH 算法,生成智能家庭网关的公钥和私钥,并将公钥发送给组网终端,智能家庭网关端将组网终端的公钥与智能家庭网关的私钥进行运算,得出共享密钥,组网终端侧与智能家庭网关端采用共享密钥加解密通讯数据,加密方式为 AES-CBC 方式;
- g) 组网终端侧收到智能家庭网关的公钥后,将智能家庭网关的公钥与组网终端的私钥进行运算, 得出共享密钥,确认密钥协商过程结束;
- h) 组网终端向智能家庭网关发起注册请求,组网终端发送自己的设备信息给智能家庭网关,请求智能家庭网关下发配置信息;
- i) 智能家庭网关回应组网终端 ACK 表示已收到该信息;
- j) 智能家庭网关与组网终端建立链路保活,开始 keepalive,验证双方是否在线。
- k) 智能家庭网关检查组网终端是否合法,如果终端合法,智能家庭网关立即下发配置信息给组网终端,如果组网终端不合法,智能家庭网关将请求手机 APP 向用户确认该组网终端是否合法,由用户判断是否下发配置信息。用户确认组网终端合法后,组网终端将加入到智能家庭网关的本地合法列表中。
- 1) 当智能家庭网关发现本地配置变化,立即发送配置文件给组网终端,组网终端用智能家庭网关发来的配置文件更新配置。
- m) 组网终端下挂设备状态上报组网终端向智能家庭网关上报当前在线下挂设备的 MAC、IP、等信息:
- n) 智能家庭网关收到组网终端上报的下挂设备信息后,回复响应消息 ACK 给组网终端,具体参数 请参见 6.1);
- o) 用户修改了智能家庭网关的 LED、WiFi 开关、WiFi 定时等其他设置参数后,也同步到组网终端:
- p) 组网终端配置完成后,并向智能家庭网关返回响应消息。

6.1.2 组网终端接入 AP 管理平台实现流程及要求

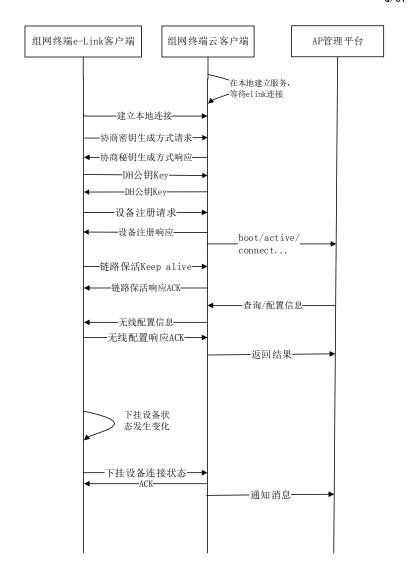


图4 组网终端与 AP 管理平台交互流程

组网终端与AP管理平台之间的通信通过云客户端实现。组网终端云客户端负责连接AP管理平台和组网终端e-Link客户端双方并进行协议转换:一方面负责和AP管理平台建立连接,接受AP管理平台的查询/配置请求;另一方面作为e-Link服务器端,和组网终端e-Link客户端进行通信,将来自AP管理平台查询/配置请求发送给e-Link客户端。

组网终端e-Link客户端需要同时连接两个e-Link服务器端:智能网关e-Link插件提供的服务器端和云客户端提供的e-Link服务器端。e-Link客户端和组网终端云客户端之间的连接、密钥协商、注册、保活等通信和交互过程与组网终端和智能家庭网关的交互相同。

组网终端e-Link客户端由AP厂商自行实现,云客户端由电信统一实现,以SDK方式提供给终端厂家进行集成。对于组网终端e-Link客户端和云客户端实现的其他要求如下:

- a) 云客户端以 UNIX domain socket 的方式,在组网终端本地提供 e-Link 服务并负责与 AP 管理平台通信。UNIX domain socket 路径为: /tmp/ctc_elinkap.sock。
- b) 组网终端启动云客户端的时候,应指定日志文件路径,并确保云客户端对日志文件有读写权限。 日志文件可以采用 ramfs 这类的内存文件系统。
- c) 组网终端上电后,应先于 e-Link 客户端启动云客户端,再建立和云客户端的内部 e-Link 连接。
- d) 组网终端应支持云客户端单独进行升级,建议参考如下方式实现:
 - ◆ 组网终端在 FLASH 中为云客户端划分单独的分区。云客户端以只读文件系统镜像的形式 发布。组网终端对客户端升级时,仅需要将新版本镜像写入分区。
 - ◆ 云客户端参数保存要求:
 - 1) 将参数以文本文件形式保存到/tmp/ctc_elinkap.json(长度小于 1Kbytes);
 - 2) 云客户端通过 e-Link 协议命令调用固件保存参数接口,将/tmp/ctc_elinkap.json 文件和固件参数一同保存;
 - 3) 固件启动时,将 ctc_elinkap.json 恢复到/tmp 目录中,再启动云客户端。
 - ◆ 当组网终端收到升级云客户端的指令时,应按以下步骤进行升级:
 - 1) 停止云客户端;
 - 2) 获取新的云客户端镜像文件;
 - 3) 检查文件完整性;
 - 4) 写入 FLASH 分区;
 - 5) 根据升级指令要求可选进行重启;
 - 6) 启动云客户端;
 - 7) 启动内部 eLink 和云客户端连接。
 - ◆ 组网终端启动时,需要检查分区内容正确性,当错误时,需要从内置的服务器地址去下载 镜像文件恢复该分区内容。
- e) 组网终端需要对云客户端提供进程守护,当发现云客户端异常退出时,需要在3秒之内重新启动云客户端。

6.2 应用终端快速接入智能家庭网关实现流程

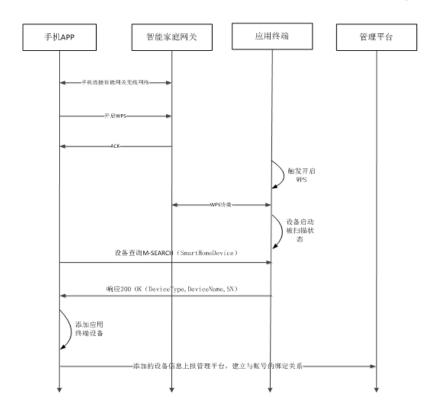


图5 智能设备发现并接入智能家庭网关流程图

流程说明:

- a) 手机 APP 通过 WiFi 接入智能家庭网关;
- b) 手机 APP 开启智能家庭网关 WPS;
- c) 智能家庭网关向手机 APP 反馈 WPS 开启状态;
- d) 根据手机 APP 提示, 触发应用终端开启 WPS;
- e) 应用终端与智能家庭网关进行 WPS 协商;
- f) 应用终端启动被发现状态;
- g) 手机 APP 启动发现应用终端动作,并获得应用终端类型、设备名称、设备厂家、序列号等信息;
- h) 手机 APP 完成对发现应用终端的添加; 手机 APP 将添加的应用终端信息上报至管理平台,管理平台保存智能家庭网关账号与应用终端的绑定关系。

7 接口定义

智能家庭网关与家庭终端的接口定义包括:智能家庭网关与组网终端的Tn接口、智能家庭网关与应用终端的Ta接口。智能家庭网关与组网终端的交互数据采用JSON协议,并定义了专用的接口数据包头、接口函数等;智能家庭网关与应用终端的交互数据采用WPS和SSDP协议,并定义了专用的接口数据格式、接口函数等。

7.1 Tn 接口——智能家庭网关与组网终端的同步接口

Tn接口交互数据采用JS0N格式,在数据发送前增加数据头部(未加密的数据,直接在数据前增加数据头部;需要加密的数据,在加密后的密文前增加数据头部),通过已建立好的TCP链接直接发送。

数据头部总共8个字节,包含两部分,标示Flag 4个字节,长度Len 4个字节。

数据头部的低4个字节为Flag, 采用big endian方式, 值为0x3f721fb5。

数据头部的高4个字节为Len,采用big endian方式,用于标识后续密文长度。

数据头部格式定义如下表:

表1 Tn 接口数据头部格式

字节	0	1	2	3	4	5	6	7
值	0x3f	0x72	0x1f	0xb5	Len[31:24]	Len[23:16]	Len[15:8]	Len[7:0]

传输方向流程,如图4:



图6 Tn 接口报文传输示意图

智能家庭网关与组网终端交互采用JSON方式,报文消息数据格式如下:

表2 发起协商秘钥生成方式

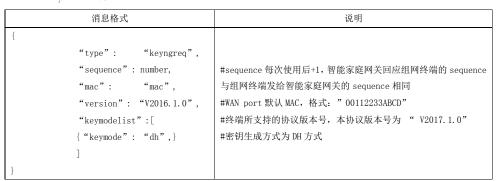


表3 回应指定密钥生成方式

消息格式	说明

```
{"type": "keyngack",#每次使用后+1,智能家庭网关回应组网终端的 sequence 与组网终端 sequence 与组网终端 sequence 相同 "mac": "mac","mac": "mac",#WAN port 默认 MAC, 格式: "00112233ABCD""keymode": "mode",#指定的密钥生成方式, mode 为"dh",
```

表4 DH 交互

消息格式	说明
{	
"type": "dh",	#每次使用后+1,智能家庭网关回应组网终端的 sequence 与组网终
"sequence": number,	端发给智能家庭网关的 sequence 相同
"mac": "mac",	#WAN port 默认 MAC,格式: "00112233ABCD"
"data":	
{	
"dh_key": "DH key",	#BASE64 编码后的 DH public key
"dh_p": "DH_P",	#BASE64 编码后的 DH P 值
"dh_g": "DH_G",	#BASE64 编码后的 DH G 值, DH 的 G 值使用十进制的 2 或者 5
}	
}	

表5 设备注册

消息格式	说明
"type": "dev_reg", "sequence": number, "mac": "mac", "data": { "vendor": "360",	#每次使用后+1 #WAN port 默认 MAC,格式: "00112233ABCD" #设备厂商 #设备型号 #软件版本 #硬件版本 #设备序列号,由平台根据终端信息统一生成,用于云客户端连接平台激活设备。 #设备 WAN IP,用于云客户端连接平台,发送心跳 #是给各厂商做自己产品介绍的链接地址 #标识该设备接入方式有线还是无线,"yes"为无线接入,"no"为12线接入

表6 链路保活

消息格式 说明

删除的内容: 有线 **删除的内容:** 无线

```
{
    "type": "keepalive",
    "sequence": number,
    "mac": "mac",
    #每次使用后+1
    #WAN port 默认 MAC, 格式: "00112233ABCD"
}
```

表7 应答

消息格式	说明
{	
"type": "ack"	,
"sequence": number,	#需要应答的报文序列号
"mac": "mac"	, #WAN port 默认 MAC,格式: "00112233ABCD"
}	

表8 配置与同步信息

在数据流的角度上,同步信息与设置信息是智能家庭网关向组网终端发送的设置请求,同步信息与设置信息共用相同的类型。

```
消息格式
                                                    说明
                                     #每次使用后+1
"type":
          "cfg",
                                     #WAN port 默认 MAC, 格式: "00112233ABCD"
"sequence": number,
"mac":
           "mac",
"status":
         {
  "WiFi" :[
                                     #WiFi mode 2.4G/5G
                                     #网关通告组网终端自己的 channel
       "radio":{
         "mode": "2.4G",
         "channel" :number,
          }
     ]
  }
"set":{
  "WiFi":[
                                     \#WiFi \mod 2.4G/5G
                                     #0 为自动
       "radio":{
         "mode": "2.4G",
         "channel": number,
                                     #txpower, 无线发射功率三个级别, 0:最高档
         "txpower": "%s"
                                     (100%); 1,中档 2:最低档。
          }
       "ap":[
                                     #虚拟 AP 的索引号,从 0 开始,0 为主 AP
```

```
"apidx": number,
         "enable": "yes",
                      "ssid",
         "ssid":
                                       #" yes" 打开, "no" 关闭该虚拟 AP。
         "key":
                      "WiFi key",
         "auth":
                     "auth mode",
                                       #认证模式, 合法的有:
         "encrypt": "encrypt mode",
                                       open, share, wpa, wpa2, wpapsk, wpa2psk, wpapsk
                                       wpa2psk,
                                       #加密模式,合法的有: none, tkip, aes, aestkip
     ]
]
```

表9 设置信息

```
消息格式
                                                              说明
     "type":
                  "cfg",
     "sequence": number,
                                               #每次使用后+1
     "mac":
                 "mac",
                                               #WAN port 默认 MAC,格式: "00112233ABCD"
     "set": {
                                               #WiFi 状态 ON/OFF, WiFiswitch 控制整个wifi
      "WiFiswitch" :{ "status" : "status" }
                                               模块,如果 WiFiswitch 为关闭(OFF),表8中
                                               apidx的 enable 字段不论是"yes"还是"no",
     "ledswitch": { "status": "status" }
                                               均为关闭。如果 WiFiswitch 总开关为开启
"WiFitimer":
                                               (ON), apidx 的字段 enable 为 "yes"时,该虚
                                               拟接口打开,为"no"时,该虚拟接口关闭。
             { "weekday" : "day",
                                               #LED 状态 ON/OFF,
             "time": "time",
                                               #周一至周日,1-7
             "enable": "enable",
                                               #执行动作的时间,例如"19:30"
                                               #0:disable 关闭, 1: enable 开启
             ]
                                               注: WiFitimer 为空表示关闭 WiFi 定时开关。示
                                               例如下:
                                                  "type":
                                                             "cfg",
                                                  "sequence": number,
                                                  "mac":
                                                             "mac",
                                                  "set": {
                                                  "WiFiswitch" :{ "status" :
                                                   "status"
```

表10 查询信息

```
消息格式
                                                                说明
"type": "get_status",
                                                #每次使用后+1
"sequence": number,
                                                #WAN port 默认 MAC, 格式: "00112233ABCD"
"mac":
            "mac",
                                                #查询信息中, "WiFi"、"WiFiswitch"、
"get": [
                                                "ledswitch"、"WiFitimer"是智能家庭网关
      { "name":" WiFi",}
                                                对组网终端查询的参数种类。
      { "name": " WiFiswitch",}
                                                # "bandsupport"是终端支持的频段(2.4G,
      { "name": "ledswitch", }
      { "name": " WiFitimer",}
                                                # "uploadspeed" 请求后的终端返回 1s 内的上
      { "name": " bandsupport",}
                                                行速率 kbps。
      { "name" : "cpurate" },
                                                #"downloadspeed" 请求后的终端返回 1s 内的
      { "name" : "memoryuserate" },
                                                下行速率 kbps。
      { "name" : "uploadspeed" },
                                                # "terminalNum"
      { "name" : "downloadspeed" },
                                                # wlanstats: 查询外置 AP 当前无线运行统计
                                                #onlineTime:设备在线时长,单位秒
      { "name" : "wlanstats" }
                                                #terminalNum:下挂终端数目#load: 查询无线信
      { "name" : " onlineTime" },
      { "name": "terminalNum"},
                                                道负载#增加 real_devinfo 和 elinkstat, 建议
                                                该接口支持未来 eLink 规范中, 无需参数的获取
      { "name" : "load" },
      { "name" : "real_devinfo" },
                                                #AP 支持该查询命令的参数以任意组合和顺序出
      { "name" : "elinkstat" },
      { "name": "neighborinfo"},
                                                #neighborinfo,查询外置 AP WLAN 频段邻居信息
      {"name": "networktype"},
                                                #networktype,接入方式,PPPOE/DHCP/STATIC
      {"name":"workmode"},
                                                #workmode, 工作模式, router/bridge/repeater
```

表11 状态返回信息

4- 社自 松	7H nH
用 尼恰八	说明

```
"type": "status",
"sequence": number,
                                                  #每次使用后+1
"mac":
          "mac",
                                                  #WAN port 默认 MAC, 格式: "00112233ABCD"
"status": {
" WiFi":[
"radio":{
"mode": "2.4G",
                                                 # WiFi mode 2.4G/5G
"channel": number,
"txpower":"%s",
"ap" :[
                                                  #虚拟 AP 的索引号,从 0 开始,0 为主 AP
"apidx":
             number
                                                  #" yes" 打开, "no" 关闭
"enable": "yes",
"ssid": "ssid",
"key": " WiFi key",
"auth" : "auth mode",
"encrypt": "encrypt mode",
                                                  #WiFi 状态 ON/OFF},
" WiFiswitch" :{ "status" : "status",
                                                  #LED 状态 ON/OFF}
"ledswitch": { "status": "status",
" WiFitimer":
                                                  #周一至周日,1-7
{ "weekday" : "day",
                                                  #执行动作的时间,例如"19:30"
"time": "time",
                                                  #0:disable 关闭,1: enable 开启
"enable": "enable",
                                                  状态信息中, "WiFi"、"WiFiswitch"、
                                                  "ledswitch"、"WiFitimer"对应指定的查询
                                                  参数种类。
" bandsupport":
                                                  # "txpower" 无线发射功率强度, "2.4G":
[ "band", "band"],
                                                  2.46 发射功率强度(0:高,1:中,2:低), "5G":
"cpurate":"%s",
                                                  5G 发射功率强度 (0: 高, 1: 中, 2: 低),
                                                  # "band" 终端支持的频段,[ "2.4G", "5G"]。
"memoryuserate": " %s",
"uploadspeed":"%s",
                                                  # "uploadspeed" 请求后的终端返回 1s 内的上
"downloadspeed": " %s",
                                                  行速率 kbps。
                                                  #"downloadspeed" 请求后的终端返回 1s 内的
"wlanstats":
                                                  下行速率 kbps。
```

```
"apidx" :number,
"ssid" :"%s",
                                                    # "wlanstats" :
                                                    #apidx": SSID 索引
"band":"%s",
"totalBytesSent":number,
                                                    #"ssid": SSID 名称
"totalBytesReceived": number,
                                                    #"band": 频段类型,取值为"2.4G"、"5G"
"totalPacketsSent": number,
                                                    #"totalBytesSent":总发送字节数
"totalPacketsReceived": number,
                                                    #"totalBytesReceived":总接收字节数
"errorsSent": number,
                                                    #"totalPacketsSent":总发送包数
                                                    #"totalPacketsReceived":总接收包数
"errorsReceived": number,
                                                    #"errorsSent":发送出错的包数
"discardPacketsSent": number,
"discardPacketsReceived": number
                                                    #"errorsReceived":接收的错误包数
                                                    #"discardPacketsSent":发送时丢弃的包数
}],
"onlineTime":"%s",
                                                    #"discardPacketsReceived":接收时丢弃的包
"terminalNum":{
"wiredNum": number,
"2.4GwirelessNum": number,
                                                    #"load"无线信道负载百分比,示例:80%。
"5GwirelessNum": number}
"channel": {
"2.4G" :number,
"5G" :number
"load": {
"2.4G":" %s",
"5G": "%s"
                                                    #下挂终端的 MAC 地址
"real_devinfo":
                                                    #下挂终端的 hostname (可选,如果存在需要上
   [
                                                    #下挂终端在线时长
"mac": "mac",
"hostname": "%s",
                                                    #下挂终端上行实时速度
"onlineTime": " onlineTime",
                                                    #下挂终端下行实时速度
"uploadspeed": "uploadspeed",
                                                    #下挂设备与组网终端的连接形式(0 有线,1 无
 "downloadspeed": "downloadspeed",
                                                    #下挂设备的 RSSI 信号(dbm)
 "connecttype": number,
                                                    #下挂终端接入频段: "2.4G"、"5G":
"band": "%s",
"rssi": "rssi",
                                                    #connectedGateway, eLink 成功连接到家庭网关
                                                    时为 yes, 反之 no
"elinkstat":{
"connectedGateway": "yes|no",
```

```
"neighborinfo":
                                                    #neighborinfo:
   [
                                                    "rfband": 频段类型,取值为"2.4G"、"5G"
                                                    "ssidname":SSID 名称
    {
"rfband": "%s",
                                                    "bssid":接口 MAC 地址
"ssidname": "%s",
                                                    "networktype":网络类型,取值"Ad-Hoc" /
"bssid": "%s",
                                                    "AP"。
                                                    "channel ":当前使用的信道
"networktype": "%s",
"channel": number,
                                                    "rssi":信号强度,单位 dBm
"rssi": "%s",
                                                    "standard":使用的 802.11 标准,取值
                                                    11a/11b/11g/11n/11ac
"standard":"%s"
                                                    #networktype,接入方式,PPPOE/DHCP/STATIC
                                                    #workmode,工作模式,router/bridge/repeater
"networktype":"%s",
"workmode":"%s",
```

表12 下挂设备状态信息

消息格式	说明
{	
"type": "dev_report",	
"sequence": number,	#每次使用后+1
"mac": "mac",	#WAN port 默认 MAC,格式: "00112233ABCD"
"dev":	
[
{	
"mac": "mac",	#下挂设备的 MAC, 格式: "00112233ABCD"
"vmac": "mac",	#下挂设备在中继设备下的虚拟 MAC,格
"connecttype": number,	式: "00112233ABCD"
}	#下挂设备与组网终端的连接形式(0 有线,1 无
]	线)
}	

表13 WAN 状态信息

DIV PALLE IN	
	说明
1 品价式	LYT. P/T

```
#每次使用后+1

"sequence": number,
"mac": "mac",
"status": {
    "ipaddr": "%s",
    "status": "%s"
}

#WAN IP, 格式 "11. 22. 33. 44"
#WAN 状态,取值: up/down/ip_changed
#AP WAN 状态变化时发送,通知云客户端是否需要和平台连接
```

表14 WPS 开关消息

消息格式	说明
<pre>{ "type": "cfg", "sequence": number, "mac": "mac", "set": { "wpsswitch":{"status": "status"} } }</pre>	#每次使用后+1 #WAN port 默认 MAC,格式:"00112233ABCD" #WPS 连接状态 ON/OFF,"NO"打开,"OFF"关闭

表15 设备升级消息

消息格式 说明	
<pre>{ "type": "cfg", "sequence": number, "mac": "mac", "set": { "upgrade":{ "downurl": "%s", "isreboot": "%s"} }</pre>	组网设备必须支持软件双备才响应升级消息,避免在升级过程中断电导致组网设备无法恢复。 #每次使用后+1 #WAN port 默认 MAC, 格式:"00112233ABCD" #downurl 软件版本文件下载地址 #isreboot 为是否立即重启(取值为:0-立即重启生效,1-等待重启生效,默认为0),

表16 设备云客户端消息

消息格式	说明
117 117. 1	V-74

```
云客户端升级步骤:
                                   1. 停止云客户端;
                                   2. 从 downurl 获取完整客户端镜像文件;
                                   3. 检查校验和是否正确,错误则不升级;
"type": "cfg",
                                   4. 写入镜像文件;
"sequence": number,
                                   5. 根据 isreboot 要求按需重启组网终端;
"mac": "mac",
                                   6. 启动云客户端;
"set": {
                                   7. 启动 eLink;
"upgrade_client":{
                                   #每次使用后+1
"downurl": "%s",
                                   #WAN port 默认 MAC, 格式:"00112233ABCD"
"chksum": "MD5Checksum"
                                   #downurl 软件版本文件下载地址
"isreboot": "%s"}
                                   #chksum: 云客户端镜像文件的 MD5, 小写 16 进制
                                   字符串(共32字符),例如:
                                   dc180f8378ba050a02cc2ba4f491ae1d
                                   #isreboot 为是否立即重启 AP (取值为: 0-立即重
                                   启生效, 1-不立即重启 AP),
```

表17 设备操作信息

消息格式	说明
{ "type": "cfg" "sequence": number, "mac": "mac", "set": { "ctrlcommand": "%s"}, }	组网设备 #每次使用后+1 #WAN port 默认 MAC, 格式:"00112233ABCD" # "reboot" / "reset" / "save" "reboot" 为设备重启, / "reset" 设备恢复 恢复出厂/ "save" 将/tmp/ctc_elinkap. json 文件和固件参数一同保存

表18 漫游配置

消息格式 说明	
---------	--

```
漫游配置接口,该功能使能后,一旦 AP 判断到
                                      STA 已经连接 AP 一段时间 (start_time),或者
                                      STA 到该 AP 的 RSSI 大于特定值( start_rssi),
                                       则启动漫游上报的判断逻辑,在 STA 到该 AP 的
                                       RSSI低于特定值时(threshold_rssi),上报RSSI
"type":
            "cfg"
                                       给智能网关。
"sequence": number,
"mac": "mac",
                                       #enable, 打开"yes"或关闭"no"漫游功能
"roaming_set": {
                                       #threshold_rssi, RSSI 上报阈值, 单位为 dBm。
"enable": "yes",
                                       漫游上报逻辑生效后,当 STA 的 RSSI 低于该值
"threshold_rssi" : xxx ,
                                       时,开始上报 STA 的 RSSI 信息
                                      # report_interval, 当 STA 的 RSSI 低于
"report_interval":xxx,
                                       threshold_rssi 后,上报 STA 信息的间隔时间,
"start_time": xxxx,
                                       单位为秒
"start_rssi": xxxx
                                       #start_time,当 STA 连接上 AP 达到该时长之后,
                                       开始启用漫游上报判断逻辑。单位为秒
                                      #start_rssi, 当新连接到 AP 的 STA 信号强于该
                                       RSSI 值时,开始启用漫游上报判断逻辑。单位
```

表19 漫游终端 RSSI 上报

消息格式	说明
	漫游终端 RSSI 上报接口。在漫游上报判断逻辑
	生效后,当STA的信号强度低于 threshold_rssi
	时,进行 RSSI 等信息上报。
"type": "cfg"	智能网关将根据上报信息判断是否让 AP 将该
"sequence": number,	STA 无线去关联一次。对于已去关联但又在 30
"mac": "mac",	秒内再次从本 AP 上线的 STA, 在当前连接断开
"roaming_report": [{	前,AP不再上报该STA的RSSI,即本次连接不
"mac":"%s",	再进行漫游。
"connect_time":xxx,	
"rssi":xxx	# mac, STA 的 MAC 地址
}]	# connect_time , STA 连接于 AP 时时长, 单位
	秒
	# rssi, STA 当前 RSSI, 单位 dBm

表20 下挂终端去关联

消息格式	说明
------	----

```
{
    "type": "deassociation",
    "sequence": number,
    "mac": "mac",
    "set":{
    "mac:["mac1"," mac2"...]
    }
}

#每次使用后+1
#WAN port 默认 MAC, 格式: "00112233ABCD"
# 终端v到该指令后,将指定 mac 地址的终端去
关联一次。
```

表21 无线信号检测

消息格式	说明	
{ "type": "getrssiinfo", "sequence": number, "mac": "mac", "get":{ "mac:["mac1"," mac2"] }	#每次使用后+1 #WAN port 默认 MAC,格式:"00112233ABCD" #返回全部地址列表中终端的 rssi 值	
}		

表22 无线信号信息返回

消息格式	说明
{	
"type": "rssiinfo",	
"sequence": number,	
"mac": "mac",	
"rssiinfo":[#每次使用后+1
{ "mac":" mac",	
"band":" band",	#WAN port 默认 MAC,格式: "00112233ABCD"
"rssi":number	#返回全部地址列表中终端的 rssi 值(单位 dbm)
}	
]	
}	

Tc接口——智能家庭网关与组网终端的同步接口

7.2 Ta 接口——智能家庭网关与应用终端的实现接口

Ta接口包括手机APP与智能家庭网关的快连接口和应用终端的设备发现接口,其中,手机APP与智能家庭网关的接口用于实现应用终端快速接入家庭智能网关,应用终端的设备发现接口用于手机APP快速发现应用终端。

7. 2. 1 手机 APP 与智能家庭网关接口

本章节涉及的接口是手机APP与家庭智能网关之间的接口,详见《中国电信智能家庭网关与智能网关及应用管理平台接口技术要求》(Q/CT 2604-2015)中的5.5及8.2章节,本接口主要采用主要其中的开启WPS、查询WPS状态、关闭WPS三个消息报文进行实现。

表23 开启 WPS

表23 开启 WPS			
网关接收消息格式	%s 填充含义	网关返回消息格式	返回值及%s 填充含义
{	SequenceId 为手机客户端动态生	WPS 开启成功	
"RPCMethod": "Post",	成,表示命令序列,网关按照请求	{	
"ID":数字,	的原值返回(8位十六进制数);	"Result":0,	
"Plugin_Name":"Plugin_	ID 为平台维护的事物 ID;	"ID":数字,	
ID",	Version 为插件的版本号(已点分	"return_Parameter":"St	
"Version":"1.0",	十进制表示,分为2段);	ring"	
"Parameter": "String"	"SSID"为 WPS 绑定的网关 SSID 编	}	Status=0 表示 WPS 开启
}	号,其中"1":表示 SSID1, "2":	return_Parameter 中的	Status-U 表示 WPS 开启 成功:
	表示	String 为	成功;
Parameter 的 String 定义	SSID2 ; "3", 表示 SSID3; "4"表	{	
为	示: SSID4; "5": 表示 SSID5; "6":	"CmdType":"WPS_ON",	
{	表示	"SequenceId":"8位16进	
"CmdType":"WPS_ON",	SSID6 ; "7", 表示 SSID7; "8"表	制数",	
"SequenceId":"8位16进	示: SSID8; 缺省为"1"。"Timeout"	"Status":"0"	
制数",	表示网关执行 WPS 过程的持续时	}	
"SSID":"网关 WIFI 编号	长,以秒为单位,如到达此时间则	WPS 开启失败	Status=1 表示 WPS 开启
<i>"</i> ,	关闭 WPS 功能, 默认 120 秒。	{	失败;
"Timeout":"%s"		"Result":0,	
}		"ID":数字,	
		"return_Parameter":"St	
		ring"	
		}	
		return_Parameter 中的	
		String 为	
		{	
		"CmdType":"WPS_ON",	
		"SequenceId":"8位16进	

	制数",	
	"Status":"1"	
	}	

表24 查询 WPS 状态

网关接收消息格式	%s 填充含义	网关返回消息格式	返回值及%s 填充含义
{ "RPCMethod":"Post", "ID":数字, "Plugin_Name":"Plug in_ID", "Version":"1.0", "Parameter":"String "	Sequence Id 为手机客户端 动态生成,表示命令序列, 网关按照请求的原值返回 (8位十六进制数); ID 为平台维护的事物 ID; Version 为插件的版本号 (已点分十进制表示,分 为2段);	WPS 状态查询成功 { "Result":0, "ID": 数字, "return_Parameter":"Strin g" } return_Parameter 中的 String 为 { "CmdType":"WPS_QY", "SequenceId":"8位16进制数 ", "Status":"0", "WPSStatus":"%s", "devInfo":"%s"	WPSStatus =0 表示已进入 WPS配置状态,但设备尚未与网关进行 WPS 交互;
Parameter 的 String 定义为 { "CmdType":"WPS_QY", "SequenceId":"8位16 进制数", "SSID":"网关WIFI编号" }	"SSID"为 WPS 绑定的网关 SSID 编号,其中"1":表示 SSID1, "2":表示 SSID2; "3",表示 SSID3; "4"表示: SSID4; "5":表示 SSID6; "6":表示 SSID6; "7",表示 SSID7; "8"表示: SSID8; 缺省为 "1"。	WPS 状态查询失败 { "Result":0, "ID": 数字, "return_Parameter":"Strin g" } return_Parameter 中的 String 为 { "CmdType":"WPS_QY", "SequenceId":"8位16进制数 ", "Status":"1" }	WPSStatus =1 表示 WPS 启动失败; WPSStatus =2 表示已有设备通过 WPS 接入网络; devInfo 表示本次通过 WPS 接入网络的设备信息。默认填写该设备的 IP 地址,并可扩展支持设备的其他信息(如设备的 MAC 地址、设备类型、设备名称等),格式要求为"IP:X. X. X. X. X. XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:

表25 关闭 WPS

网关接收消息格式	%s 填充含义	网关返回消息格式	返回值及%s 填充含 义
{ "RPCMethod":"Post", "ID":数字, "Plugin_Name":"Plugin _ID", "Version":"1.0", "Parameter":"String" }	SequenceId 为手机客户端动态 生成,表示命令序列,网关按照 请求的原值返回(8 位十六进制 数); ID 为平台维护的事物 ID; Version 为插件的版本号(已点 分十进制表示,分为 2 段);	WPS 美闭成功 {	Status=0 表示 WPS 关闭成功:
Parameter 的 String 定 义为 { "CmdType":"WPS_OFF", "SequenceId":"8 位 16 进制数", "SSID":"网关 WIFI 编号 " }	"SSID"为 WPS 绑定的网关 SSID 编号,其中"1":表示 SSID1,"2": 表示 SSID2; "3",表示 SSID3; "4" 表示: SSID4; "5":表示 SSID5, "6":表示 SSID6; "7",表示 SSID7; "8" 表示: SSID8; 缺省为"1"。	WPS 美闭失败 { "Result":0, "ID": 数字, "return_Parameter":"String" } return_Parameter 中的 String 为 { "CmdType":"WPS_OFF", "SequenceId":"8 位 16 进制数", "Status":"1" }	Status=1 表示 WPS 关闭失败;

7.2.2 应用终端的设备发现接口

本章节涉及的设备发现接口是由简单服务发现协议SSDP(Simple Service Discovery Protocol)来定义实现的。SSDP常见的协议请求消息有两种类型,第一种是服务通知,设备和服务使用此类通知消息声明自己存在;第二种是查询请求,协议客户端(本规范中手机APP)用此请求查询某种类型的设备和服务。本标准是对SSDP协议的查询请求消息进行了约定,其协议消息格式如下:

表26 设备查询消息 M-SEARCH

消息类型	消息格式	说明
M-SEARCH	M-SEARCH * HTTP/1.1 HOST: 239.255.255.250:1900 MAN: "ssdp:discover" MX: seconds to delay response	ST: 本标准统一定义为: SmartHomeDevice。 其它字段消息采用标准的 SSDP 协议,本标准不做特殊要求。

消息类型	消息格式	说明
	ST: SmartHomeDevice	

表27 设备查询请求响应 200 OK 消息格式:

消息类型	消息格式	说明		
200 OK	HTTP/1.1 200 OK	1)	LOCATION: 统一定义格式为: http://设备类	
	CACHE-CONTROL: max-age = seconds until		型/设备名称/设备厂家/设备序列号;	
	advertisement expires	2)	ST:与 M-RESERCH 的保持一致,统一定义为:	
	DATE: when reponse was generated		SmartHomeDevice.	
	EXT:	:	其他字段消息不做特殊要求。	
	LOCATION:		本标准涉及的设备的定义需遵循行业相关应用	
	http://DeviceType/DeviceName/Manufac	终端类型及名称定义。		
	ture/SN	详见:		
	SERVER: OS/Version UPNP/1.0	1)	SSDP 协议标准:	
	product/version		http://tools.ietf.org/html/draft-cai-ssd	
	ST: SmartHomeDevice		p-v1-03	
	USN: advertisement UUID	2)	UPnP 协议框架:	
			http://www.upnp.org/specs/arch/UPnP-arch	
			-DeviceArchitecture-v1.0.pdf	

附 录 A 组网终端状态变迁图

组网终端在与家庭智能网关连接的过程中,它的状态是不断变化的,为了便于组网终端的无线配置自同步功能开发,组网终端在与家庭智能网关连接过程中的状态变迁,可参考下图:

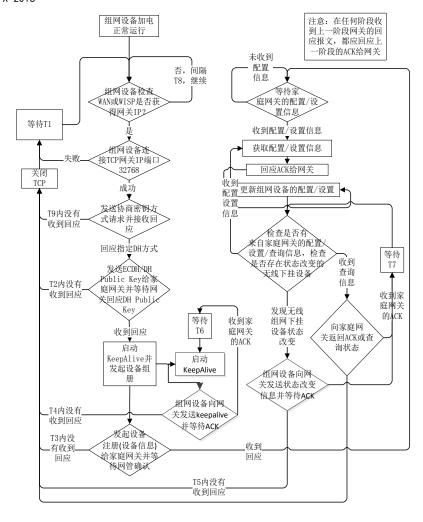


图7 组网终端状态变迁图

注3:

- a) 组网终端虚拟 AP 的个数及配置由智能家庭网关指定,组网终端不再打开配置信息以外的虚拟 AP;
- b) 超时时间: T1=10s, T2=10s, T3=20s, T4=20s, T₄=5s, T6=5s, T7=5s, T8=2s, T9=5s。