

# APRUS Lua-TCP 转发配置说明

## 概述

本章主要概述 APRUS Lua TCP 转发的相关配置说明，forward TCP 模块是智物联针对客户和市场的需求独立开发的一种解决方案。例如：当多个控制、采集设备都需要组态屏的支持，而组态屏只支持一个 TCP 连接时，那么就需要 forward TCP 模块帮忙转发、处理多个控制、采集设备的请求，达到资源共享的目的。forward TCP 模块是 APRUS Lua 的一部分，主要负责 TCP 数据转发，支持缓存模式：可以记录历史、自动采集并刷新结果，提高数据获取效率。客户只需简单配置 config.lua 就可以同时实现数据采集和转发功能。

## 1 config.lua

config.lua 一般需要配置两大部分：**适配器参数**(AprusX 段)、**TCP 转发配置段参数**(ForwardTCP 段)；其中 **TCP 转发配置段参数**又细分为**服务侧配置**(dataServer)和**转发侧配置**(forwardServer)。

config.lua 示例：

```
return
{
    APRUS X={
        ipmode="none",
        inet_addr="192.168.123.234",
        netmask="255.255.255.0",
        luaver="V00.R",
        devinfo="ForwardTCPDev",
    },
    Forward={
        dataServer={
            ifname="eth1",
            ip="192.168.123.44",
            port=502,
            type="modbusTCP",
            timeout=3000,
            query_delay = 5,
            query_TimeOut = 5,
            replay_TimeOut = 2,
        },
        forwardServer={
            ifname="eth0",
            ip="192.168.123.88",
            port = 8502,
        },
    },
}
```

}

## 1.1 适配器参数：AprusX 字段

适配器参数的功能如下表所示：

参数	值	说明
ipmode	" auto"/" manual"/" none"	APRUS 设备 IP 地址获取方式:自动/手动/无
inet_addr	" 192.168.123.234"	APRUS 设备 IP 地址；需要与 dataServer 的 IP 地址配置为同一网段
netmask	" 255.255.255.0"	子网掩码
luaVer	"V00.R"	APRUS Lua 版本信息，根据实际自行填写
devinfo	" ForwardTCP"	与 APRUS 对接设备，可根据需求自行填写

## 1.2 TCP 转发配置段参数：服务侧和转发侧

在之前的例子中，组态屏就是服务侧，多个模块就是转发侧。

**服务侧**是指提供服务的服务器。服务侧有且只有一个，它是所有转发侧的共享资源。服务侧负责处理来自转发侧的请求，并返回相应的结果。

**转发侧**是由 APRUS 提供的 TCP 服务器。转发侧监听并接受转发端口的连接请求，可同时提供多个 TCP 连接，为多个客户端提供服务。那些需要使用服务侧资源的客户端可以连接到这个服务器，通过向转发服务器发送请求来获取服务侧资源，这些请求可能包括数据获取、数据更新、事务处理等操作。

### 1.2.1 服务侧（dataServer）

服务侧参数功能如下表所示：

参数	值	说明
ifname	" eth1"	绑定硬件网口（可不填，默认为 eth1）
ip	" 192.168.123.44 "	被转发设备的 IP 地址（必填）
port	502	被转发设备的网络端口（必填）
type	" modbusTCP"	通信协议适配（可不填，默认为一般模式 modbusTCP 模式，在对比查询命令的时候会忽略通信包中前 4 字节数据。）
timeout	3000	通信超时（可不填，默认为 3000ms）
query_delay	5	缓存模式下 查询通信间隔 （可不填，默认为 10uS。如果快速读取导致服务侧通信故障，请

		增加此值)
query_TimeOut	5	缓存模式下 查询过期时长 (可不填, 默认为 20S。如果缓存的命令在规定时长内没有被再次使用, 则从缓存中删除查询和应答数据)
replay_TimeOut	2	缓存模式下 应答过期时长 (可不填, 默认为 3S。如果缓存中的应答采集时间与当前时间差 3 秒以上, 则不使用缓存应答数据)

## 1.2.2 转发侧 (forwardServer)

转发侧节点参数功能如下表所示:

参数	值	说明
ifname	"eth1"	绑定硬件网口 (可不填, 默认为 eth0)
ip	"192.168.123.44"	转发侧设备的 IP 地址 (可不填, 前提是对应网口已有 IP 地址)
port	502	转发侧设备的网络端口 (必填, 且不可与被转发设备的网络端口一致)

## 2 APRUS.lua

### 2.1 aprus.Lua 配置样例

```

opackage.cpath="./?.so"

package.path="./?.lua"

cjson = require "cjson"

config = require "config"

require "forwardTCP"

function init_forwardTCP()

    fwdtcpObj = forwardTCP.new("forwardTCP")

    dataCfg = cjson.encode(config.ForwardTCP.dataServer)

```

```
forwardCfg = cJSON.encode(config.ForwardTCP.forwardServer)

forwardTCP.config(fwdtcpObj, dataCfg, forwardCfg)

forwardTCP.cachemode(fwdtcpObj, 1)      --启用缓存模式

forwardTCP.run(fwdtcpObj)

end

function start()

    init_forwardTCP()

    while true do

    end

end

start()
```

- 1、代码中"--"后面的部分为注释语句，仅供解释代码意图，不对程序流程有所干预。
  - 2、在源码的开头部分调用需要的模块如：`require "forwardTCP"`
  - 3、所有函数以 `function` 开头，以跟它对齐的 `end` 结尾。代码使用 `TAB` 缩进对齐。
- 例：

```
function init_forwardTCP()

    fwdtcpObj = forwardTCP.new("forwardTCP")

    dataCfg = cJSON.encode(config.ForwardTCP.dataServer)

    forwardCfg = cJSON.encode(config.ForwardTCP.forwardServer)

    forwardTCP.config(fwdtcpObj, dataCfg, forwardCfg)

    forwardTCP.cachemode(fwdtcpObj, 1)      --启用缓存模式

    forwardTCP.run(fwdtcpObj)

end
```

#### 4、初始化 `init_forwardTCP()` 函数解析

--函数定义：函数名 `init`，无参数

```
function init_forwardTCP ()

    --调用 forward 模块的功能 forwardTCP.new () 创建一个新的模块实体对象 fwdtcpObj 。

    fwdtcpObj = forwardTCP.new("forwardTCP")

    --调用 forward 模块的功能 forwardTCP.config (), 为新创建的模块实体对象进行配置, 配置
    参数为 config.lua 中的通信端参数

    forwardTCP.config(fwdtcpObj, dataCfg, forwardCfg)

    --启用缓存模式

    forwardTCP.cachemode(fwdtcpObj, 1)

    --调用 forwardTCP.run() 功能, 运行 forwardTCP 的实体对象 fwdtcpObj

    forwardTCP.run(fwdtcpObj)

--初始化函数 init_forwardTCP () 结束

end
```

## 2.2 forwardTCP 函数功能介绍:

### 2.2.1 forwardTCP.new

**功能:** 创建 forwardTCP 实例

**示例:** fwdtcpObj = forwardTCP.new("forwardTCP")

参数	值	说明
"forwardTCP"		模块运行实例的名称

### 2.2.2 forwardTCP.config

**功能:** 配置 forwardTCP 实例

**示例:** forwardTCP.config(fwdtcpObj, dataCfg, forwardCfg)

参数	值	说明
fwdtcpObj		forwardTCP 实例的返回值
dataCfg	cjson.encode(config.ForwardTCP.dataServer)	服务侧参数 (详见 config.lua)

forwardCfg	cjson.encode(config.ForwardTCP.forwardServer)	转发侧参数（详见 config.lua）
------------	---	----------------------

### 2.2.3 forwardTCP.cachemode

**功能：**开启/关闭缓存模式。缓存模式开启后可提高数据获取效率，减轻服务压力，确保系统稳定高效。

**示例：**forwardTCP.cachemode(fwdtcpObj, 1)

参数	值	说明
fwdtcpObj		forwardTCP 实例的返回值
enable	0 / 1	缓存模式开关 1：开启 0：关闭

### 2.2.4 forwardTCP.run

**功能：**创建 TCPforward 启动实例

**示例：**obj = forwardTCP.run(fwdtcpObj)

参数	值	说明
fwdtcpObj		forwardTCP 实例的返回值