

# TSN 集中控制器设计

## （版本 1.1）

OpenTSN 开源项目组

2021 年 1 月

## 版本历史

[illegible]

## 目录

1、设计目标.....	4
2、总体设计.....	4
2.1 系统结构.....	4
2.2 实现架构.....	5
2.3 工作流程.....	6
2.3.1 控制器流程图.....	6
3、详细设计.....	错误!未定义书签。
3.1 网络初始化.....	错误!未定义书签。
3.1.1 整体架构.....	错误!未定义书签。
3.1.2 整体流程.....	错误!未定义书签。
3.1.3 数据结构.....	错误!未定义书签。
3.1.4 关键函数定义.....	错误!未定义书签。
3.2 状态监测.....	错误!未定义书签。
3.2.1 整体架构.....	错误!未定义书签。
3.2.2 整体流程.....	错误!未定义书签。
3.2.3 数据结构.....	错误!未定义书签。
3.2.4 关键函数定义.....	错误!未定义书签。
3.3 ARP 代理.....	错误!未定义书签。
3.2.1ARP 设计.....	错误!未定义书签。
3.2.2 整体流程.....	错误!未定义书签。
3.2.3 数据结构.....	错误!未定义书签。
3.2.4 关键函数定义.....	错误!未定义书签。
附录 1 TSNtag 格式.....	9
附录 2 命名说明.....	10
附录 3 arp 表 xml 文本.....	10
附录 4 修改记录.....	11

## 1、设计目标

为了避免时间敏感网络中每个交换机都需要一个控制器进行控制的情况，提出 TSN 集中控制器，该控制器能够对时间敏感网络中所有交换机进行配置（包括对 HCP 和枫林 TSN 的所有配置），以及实现软硬件协同的时间同步，ARP 响应等，并进行网络监测。该 TSNLight2.1 版本可以在 OpenTSN2.0 以及 OpenTSN3.0 中使用。

## 2、总体设计

### 2.1 系统结构

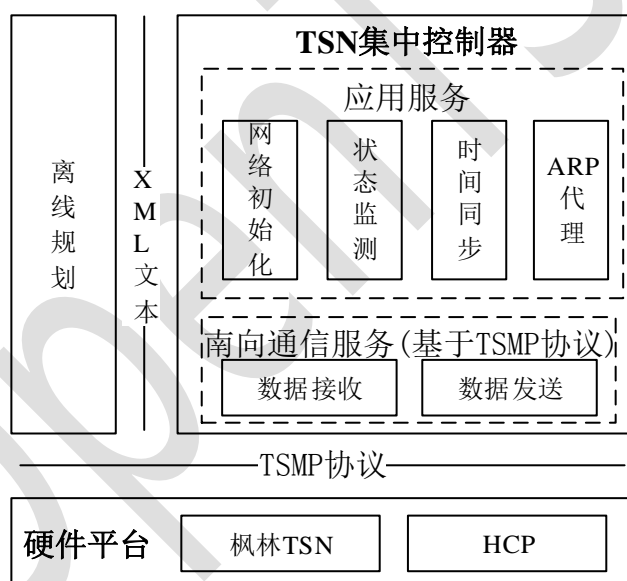


图 2-1 TSN 集中控制器系统结构

TSN 集中控制器的系统架构如上图所示，主要应用包括网络初始化（net\_init）、状态检测（state\_monitor）、时间同步（PTP）和 ARP 代理（arp\_proxy）。其中应用与硬件平台交互使用 TSMP 协议，离线规划与控制器交互使用 XML 文本。离线规划作为 TSN 集中控制器的外围工具，用于生成配置信息。

## 2.2 实现架构

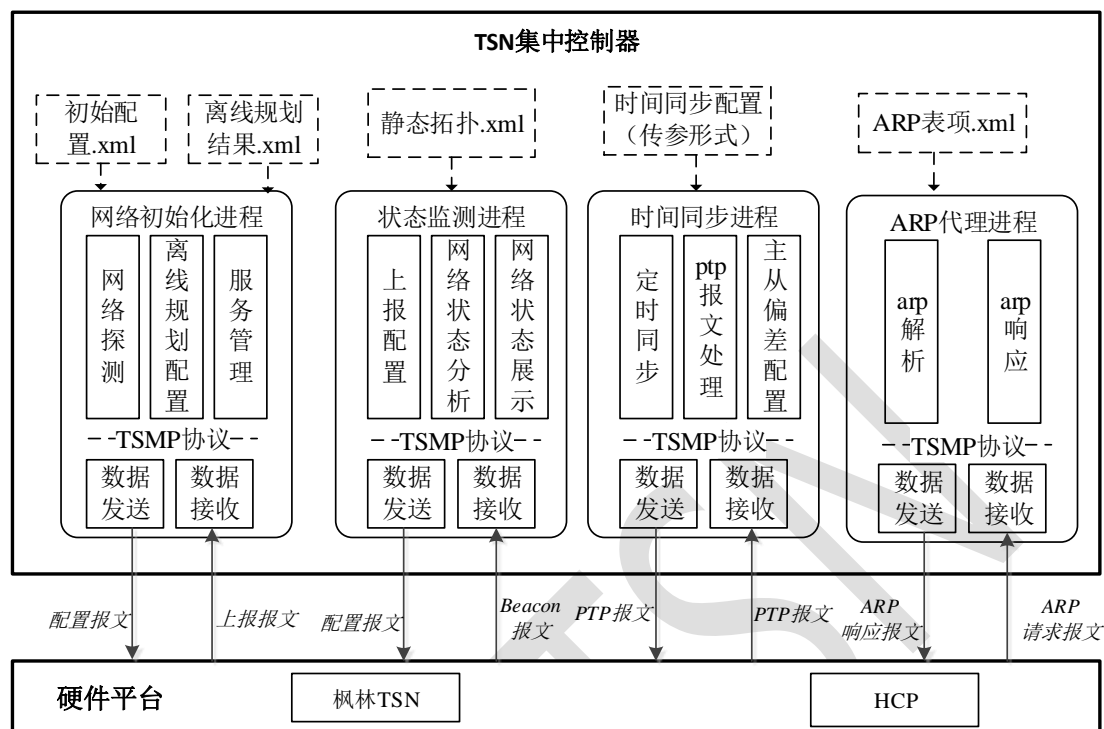


图 2-2 TSN 集中控制器实现架构图

TSN 控制器的实现架构如上图所示，该架构采用多进程的形式，每个应用为一个独立的进程，方便各个应用进行独立开发。应用进程通过 TSMP 协议与硬件平台交互，通过 XML 文本与外部应用进行交互。在控制器启动时，网络初始化进程为开始进程，需要首先运行，然后在网络初始化进程的不同阶段开启其他进程。

网络初始化进程：主要完成网络探测和网络配置的功能。网络探测是根据初始配置文本完成初始配置，检测链路是否通畅（对节点进行初始配置，如果能接收到节点上报报文，表明该节点没有问题）；离线规划配置表示根据细线规划的结果，对硬件表项和寄存器记性配置；服务管理主要功能为开启和关闭相应服务。

状态监测进程：主要功能是根据上报报文判断网络的状态。上报

配置用于对交换机进行上报管理，包括开启或关闭上报，选择不同的上报类型；网络状态分析用于解析上报报文，获取各个交换机的统计信息和状态，并分析网络状态；网络状态展示表示将目前的网络状态信息展示在界面上。

时间同步进程：完成软硬件协同时间同步功能。定时同步用于定时发送 sync 报文，用于设定同步周期；ptp 报文处理用于对接收到的 ptp 报文进行解析，获取时间戳和透明时钟，并构造响应的 ptp 报文；补偿配置首先需要根据时间戳和透明时钟计算出主从时间的偏差 offset，然后构建配置报文把 offset 配置到硬件中。

ARP 代理进程：主要功能是响应 ARP 请求报文。ARP 解析用于对接收到 ARP 请求报文进行解析，获取 IP 地址。ARP 响应则用于查找从文本中获取的 ARP 表项，构造 ARP 响应报文。

## 2.3 工作流程

### 2.3.1 控制器流程图

控制器的流程从网络初始化进程开始，然后由网络初始化进程开启和关闭其他进程。

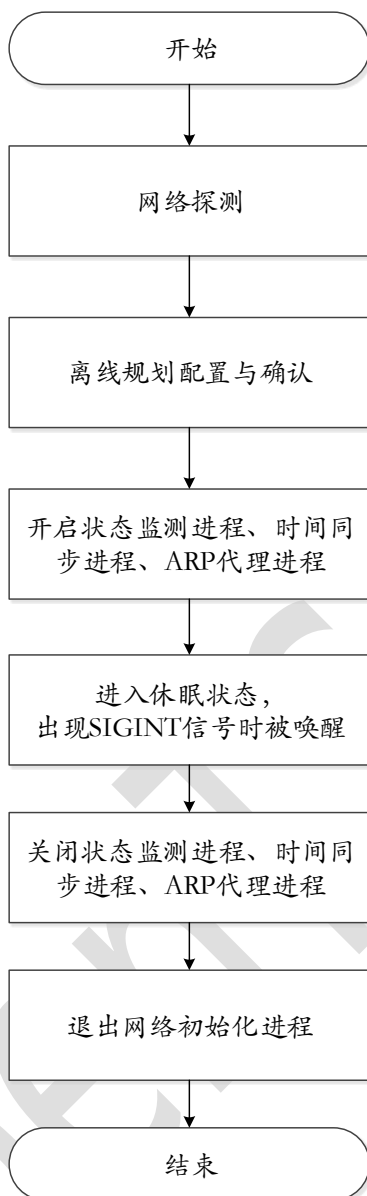


图 2-3 控制器流程图

- (1) 网络探测：控制器首先根据初始配置文本对各个交换机进行进行初始配置，并根据是否接收到各交换机的上报报文确定网络状态，完成网络探测的功能。
- (2) 离线规划配置与确认：根据离线规划的结果，从文本中读取离线配置信息，按照生成的配置内容和配置顺序配置所有节点。
- (3) 开启状态监测进程、时间同步进程、ARP 代理进程：在完

成网络初始化后，分别开启各个应用进程。

- (4) 进入休眠状态，出现 SIGINT 信号时被唤醒：开启其他应用进程后休眠该进程，等待 SIGINT 信号（Ctrl+c）被唤醒。
- (5) 关闭状态监测进程、时间同步进程、ARP 代理进程：唤醒后关闭所有的应用进程。
- (6) 退出网络初始化进程：最后退出网络初始化进程。



## 附录 1 TSNTag 格式

表附-1 时间同步报文的 TSNTag

位置	名称	位置	描述
3	Flow type	[47:45]	流类型。 100: 同步报文 （其他报文的格式如下表）
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换节点命中同一条表项。
12	Reserve	[30:19]	保留
19	Rx_timestamps	[18:0]	芯片接收到时间同步报文的本地时间信息，用于芯片发送报文时计算透明时钟。

表附-2 非时间同步报文的 TSNTag

位置	名称	位置	描述
3	Flow type	[47:45]	流类型。000: TS 分组    001: TS 分组    010: TS 分组  011: RC 分组    100: 时间同步报文分组 101: NMAC 分组    110: BE 分组  111: 需重组分组（时间同步报文的格式如上表）  <b>目前控制器使用到 TSMP 头的流类型为 110: BE 分组</b>
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换节点命中同一条表项。
16	Seq id	[30:15]	用于标识每条流中报文的序列号， <b>控制器 TSMP 头未使用，保持默认值 0</b>
1	Frag flag	[14]	用于标识分片后的尾。0: 分片后的中间报文 1: 尾拍  <b>控制器 TSMP 头未使用，保持默认值 0</b>

4	Frag ID	[13:10]	用于表示当前分片报文在原报文中的分片序号  控制器 TSMP 头未使用，保持默认值 0
5	inject addr	[9:5]	TS 流在源端等待发送调度时缓存地址  控制器 TSMP 头未使用，保持默认值 0
5	submit addr	[4:0]	TS 流在终端等待接收调度时缓存地址  控制器 TSMP 头未使用，保持默认值

## 附录 2 命名说明

- 1、接口类型：主机口、网络（网络口包含两种模式，协作模式、非协作模式，）

表附-3 接口类型

接口	主机口	与 HCP 连接的端口
	网络口	网络口分为两种工作模式 1、协作模式：只接收映射过的报文 2、非协作模式：接收任何格式的报文，经过主机进行映射。

- 2、报文类型

表附-4 报文类型

报文类型	标准以太网报文	标准的以太网报文
	TSN 报文	在 TSN 网络中传输的报文，包含三种类型的报文  1、ST 报文：时间敏感报文 2、RC 报文：带宽预约报文 3、BE 报文：尽力转发报文

## 附录 3 arp 表 xml 文本

表附-5 arp 表 xml 文本

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<nodes>

<arp_table>

<entry IP="192.168.1.56" MAC="c0:00:80:00:00:00" /> //imac=0x01

<entry IP="192.168.1.57" MAC="c0:01:00:00:00:00" /> //imac=0x02

</arp_table>

</nodes>
```

附录 4 修改记录

表附-6 修改记录

日期	修改内容	修改人
2020.11.20	<p>1、修改实现架构图中，离线规划结果只有网络初始化进程使用，状态监测不再需要，因为网络状态展示由另外一个进程实现，不需要在状态监测模块展示。</p> <p>2、时间同步模块不再需要读取文本，使用主函数传递参数的形式传递到时间同步进程中</p> <p>3、修改初始配置的 xml 文本，在根节点中增加 sync_period、master_imac、sync_imac，并在定义该三个参数的全局变量，方便给时间同步进程传递参数。</p> <p>4、把初始配置文本中的 port_type 修改为 port_mode，port_mode 使用 bitmap 的形式，0 代表协作模式，只接收映射过的报文，1 代表非协作模式，可以接收所有报文。</p> <p>5、修改表项配置的数据结构，由每次读取 32 条转换成每次读取一类表项，并把所有的表项都存储在数据结构中，存储的数组大小由 32 修改为 16384（转发表最大为 16384 条），</p> <p>6、删除在配置时再存储一个 16bit 的配置内容用于比较，修改后的方案为</p>	李军帅

	<p>把上报报文的位宽（16bit）先进行转换（转换成 32bit），然后再与存储的表项进行比较</p> <p>7、静态拓扑的 xml 文本中，增加 <code>remote_device</code> 的类型属性，表示远端连接的是控制器、交换机还是终端，把远端的 <code>imac</code> 修改为 <code>addr</code>，如果远端设备为控制器和交换机，则 <code>addr</code> 表示远端的 <code>imac</code>，如果为终端，则表示标签号。并修改对应的数据结构。</p> <p>8、合并 ARP 代理文档</p>	
--	---	--