

# TSN-CNC 软件设计文档

(V1.0)



*Open***TSN**

## 一、设计目标

本课题旨在通过设计 TSN 系统对 TSN 管理架构和关键技术进行实现和验证。TSN 系统分为数据平面和控制平面两部分。数据平面的主要功能包括时钟同步，分组交换和分组调度。这些功能拟基于 FPGA 硬件实现以保证精确性。控制平面的主要功能包括拓扑管理，资源管理和分配等功能。拟采用集中式的管理架构对分布式节点进行配置和管理，从而简化管理难度，提升收敛速度。

## 二、总体设计

### 2.1 整体架构

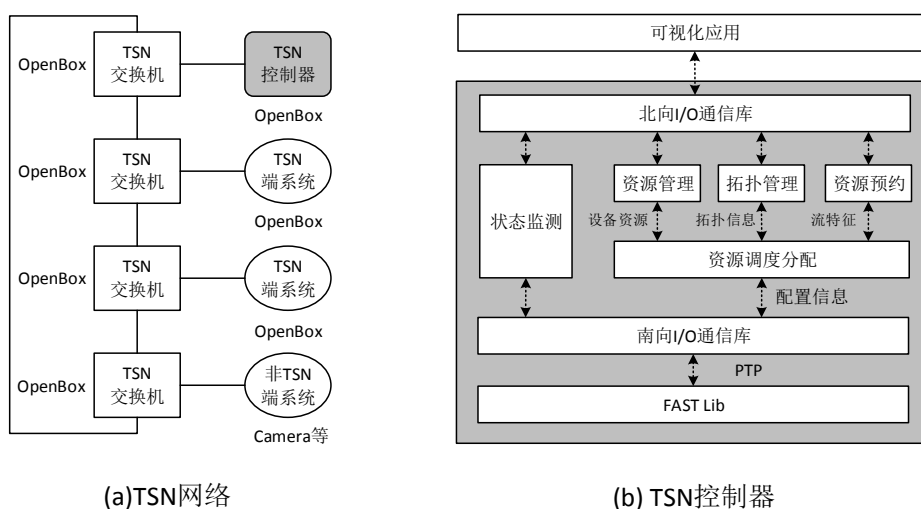


图 1 整体架构

TSN 网络的部署结构如图 1(a)所示，由 TSN 交换机构建环形拓扑。每个 TSN 交换机上有三个端口，其中两个端口用于 TSN 交换机之间互联进行数据转发，另一个端口用于连接端设备。端设备包括 TSN 端系统，非 TSN 端系统和 TSN 控制器。在本项目中，TSN 网络通过 TSN 控制器对整个网络进行集中式控制管理。一台 TSN 交换机专门连接 TSN 控制器，其中 TSN 控制器在基于 CPU/FPGA 架构的 Openbox 设备上实现。FPGA 用于对从 TSN 交换机接收的 PTP 报文标记时间戳。CPU 上软件系统用于根据资源状态信息和流特征进行预先调度，以及状态信息的收集和上报。

TSN 控制器架构如图 1(b)所示，主要包括南向 I/O 通信库，北向 I/O 通信库，状态监测模块，资源调度分配模块，资源管理模块，拓扑管理模块，资源预约模块。

- **南向 I/O 通信库：**用于与 TSN 交换机和 TSN 端系统之间的通信，实现网络配置和状态信息获取。底层调用 FAST Lib 与 FPGA 进行通信。
- **北向 I/O 通信库：**实现与北向可视化应用通信，提供可视化信息。
- **资源管理模块：**用于存储交换节点的带宽、队列长度、控制信息等资源信息。
- **拓扑管理模块：**用于存储交换节点和端系统的地址，标识号和连接关系等信息。
- **资源预约模块：**用于存储当前 TSN 网络中需要预约资源的流量特征，主要包括时间敏感流和带宽约束流。
- **资源调度分配模块：**根据全局资源和拓扑信息对预约资源的流量进行资源调度

与分配。目前主要采取离线模式进行预先资源规划。

- **状态监测模块:** 对包含可视化信息的报文进行解析, 并将可视化信息进行存储。

## 2.2 整体处理流程

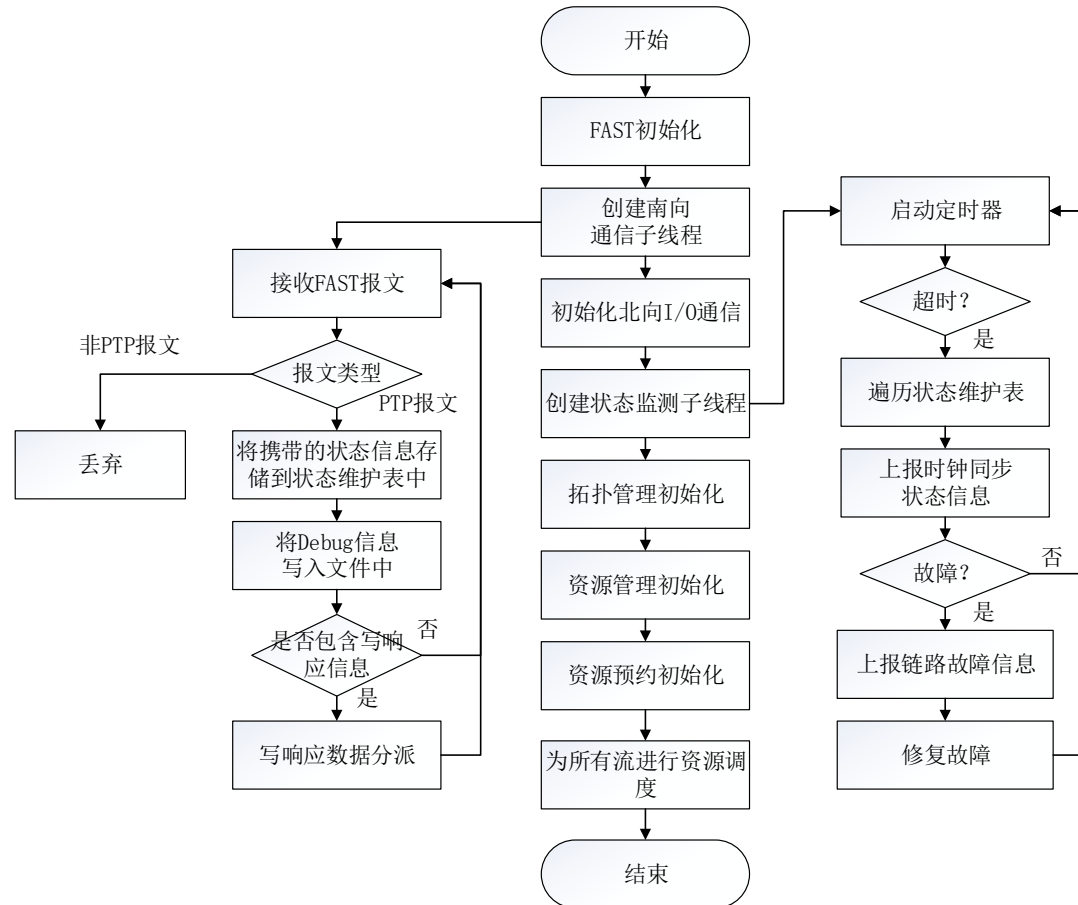


图 2 整体处理流程

TSN 控制器采用单进程多线程架构, 包括主线程, 南向通信子线程和北向通信子线程。

- **主线程**

- (1) 初始化 FAST 平台; 由于是在 openbox\_s4 中运行, 因此需要初始化 FAST 平台, 以便使用 FAST 结构。
- (2) 初始化北向 I/O 通信库; 该部分主要是把信息 (时间同步信息, 链路的状态等) 传送给可视化应用, 提供与可视化应用通信的接口函数。
- (3) 初始化拓扑管理模块, 从文本中读取静态拓扑信息, 以链表的方式把交换机与交换机之间连接起来, 并且每个节点都包含该交换机和端设备的信息, 并根据文本信息配置交换机连接的端设备 MAC 地址。
- (4) 初始化资源管理模块, 包括对设备资源 (带宽和 CQF 队列) 和配置信息的初始化, 对每个设备的资源和配置用宏定义的方式进行初始化, 初始化时每个交换机除了连接的端系统的 MAC 地址不同, 其他都是相同的。
- (5) 初始化资源预约模块, 从文本中读取用户提供的流特征信息保存在资源预约表中, 主要包括 TSN 单播流和带宽约束单播流的特征。

- (6) 为所有的 TSN 流进行资源调度和配置,包括读取所有的 TSN 单播流特征,并计算出每个流的路径,然后对所有的 TSN 流进行排序,再进行时隙调度,调度成功后进行资源配置和更新设备资源。
  - (7) 为所有的带宽预约流进行资源调度和配置,包括读取所有的带宽预约流特征,并计算出每个流的路径,然后对所有的带宽预约流进行排序,再进行带宽调度,调度成功后进行资源配置和更新设备资源。。
- 南向通信子线程
    - (1) 该线程是一个接收线程,用于接收 FAST 报文;
    - (2) 解析报文类型,目前只有 PTP 报文。如果是非 PTP 报文则丢弃,执行(1);对于 PTP 报文,解析并提取时钟同步和链路状态信息存储到状态监测模块的状态维护表中。
    - (3) 将 PTP 中的 Debug 信息(包括时间戳,配置信息和状态信息)写入文件中。
    - (4) 解析 PTP 报文中配置信息标志位是否置位。如果置位,说明包含写响应数据,则分发响应信息,需要把接收到的报文中的配置信息写入到映射的地址中,用于发送端读取该信息并确定配置是否成功,如果没有置位,则执行(1)。
  - 状态监测子线程
    - (1) 启动定时器,如果超时则说明定时时间到,执行(2);
    - (2) 遍历状态维护表,读取到每个设备的更新的时间同步信息,构造并上传时钟同步信息;
    - (3) 根据状态维护表来判断是否存在故障链路,如果有一个或多个设备没有上报时间同步信息,说明链路出现了故障,并根据拓扑查找到出现故障链路的地方。如果出现故障,执行(4);如果不存在,执行(1);
    - (4) 构造并上传链路故障报文,配置设备的转发方向,进行链路故障修复,执行(1);

## 三、详细设计

### 3.1 南向通信 I/O 库

#### 3.1.1 功能描述

南向通信 I/O 库主要提供控制器和交换机通信功能,主要包括南向协议报文和报文 I/O 操作两部分。

- (1) **南向协议报文:** 报文目前只有 PTP 报文,分为上行报文和下行报文。
  - a) **下行报文:** 指控制器下发给交换机和端系统设备的 PTP 报文,控制器通过发送包含写请求的 PTP 报文对交换机和端系统设备进行配置,目前对交换机的配置包括报文转发方向,令牌桶速率和主机 MAC 地址。交换机在收到 PTP 报文后,直接将整个配置信息写入对应的地址中覆盖原来的数据。
  - b) **上行报文:** 指交换机和端系统设备会定期向控制器上报 PTP 报文,携带配置信息,状态信息和时钟同步。配置信息是当前交换机中所有可配置地址中数值信息。状态信息是交换机中端口和模块中各类计数器信息,为可读信息。时间同步信息携带在整个传输过程中累积的透明时钟,用于计算主

时钟和从时钟的时间偏差。当交换机收到包含写请求的 PTP 报文后，在下一个交换机上报的 PTP 报文中的控制信息标志位会被置位，表明该报文中配置信息为对写请求的响应，用于软件判断是否正确写入。

- (2) **报文 IO 操作：**包括对 FAST 报文的接收处理，该函数作为 FAST lib 中的回调函数进行注册。构造包含写请求的 PTP 报文发送，其中 PTP 报文会封装 FAST 头部。对 PTP 报文进行处理。

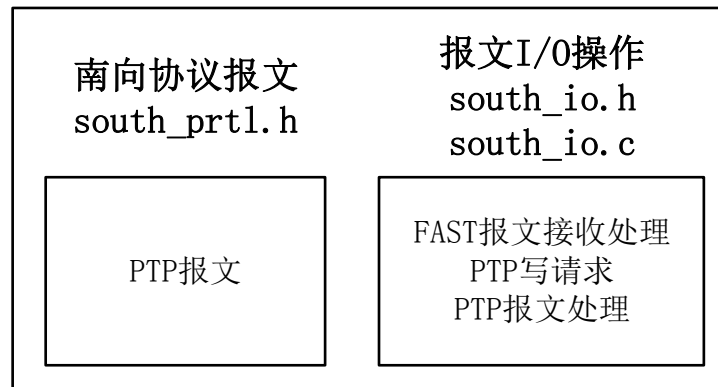


图 3 南向通信模块

### 3.1.2 数据结构

南向接口的数据结构只有 PTP 报文，主要携带对交换机配置信息，状态信息和透明时钟信息。目前对交换机的配置信息如表 1 所示，可读的状态信息如表 2 所示。

表 1 PTP 中配置信息

交换机	产生令牌的速率 rate
	报文发送方向 direction
	主机的 host_mac 地址

表 2 PTP 中状态信息

交换机	esw_pkt_in_cnt	进入 ESW 模块的分组计数器
	esw_pkt_out_cnt	ESW 输出模块的分组计数器
	local_mac_addr	TSN 本地 MAC 地址，只有 8 位软件可读
	buf_id_used_cnt	Bufrm 中所使用的 ID 计数器
	pad	填充
	eos_md_in_cnt	进入 EOS 模块的元数据计数器
	eos_md_out_cnt	EOS 模块输出的元数据计数器
	eos_q0rest_cnt	EOS 模块 Q0 队列的剩余长度计数器
	eos_q1rest_cnt	EOS 模块 Q1 队列的剩余长度计数器
	eos_q2rest_cnt	EOS 模块 Q2 队列的剩余长度计数器
	eos_q3rest_cnt	EOS 模块 Q3 队列的剩余长度计数器
	time_slot_flag	奇偶时间槽，奇数时间槽位 1，偶数为 0
	pad	填充
	goe_pkt_in_cnt	进入 GOE 模块的分组计数器

	goe_port0_out_cnt	GOE 模块往 0 口输出的分组计数器
	goe_port1_out_cnt	GOE 模块往 1 口输出的分组计数器
	goe_discard_cnt	GOE 模块丢弃的分组计数器

### (1) PTP 报文

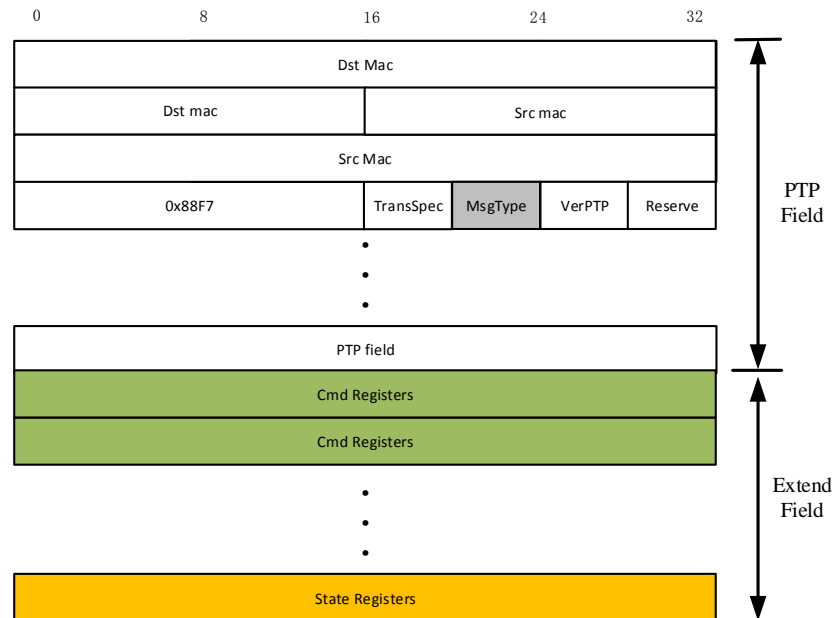


图 4 PTP 报文格式

表 3 配置信息结构体

struct config_info		
字段	位宽	含义
host_mac[6]	8	端系统 MAC 地址
direction	1	报文的转发方向，0 为 0 口转发，1 为 1 口转发
pad0	15	填充
tb_rate	32	产生令牌的速率 rate
pad1	32	填充

表 4 状态信息结构体

struct status_info		
字段	位宽	含义
esw_pkt_in_cnt	64	进入 ESW 模块的分组计数器
esw_pkt_out_cnt	64	ESW 输出模块的分组计数器
local_mac_addr	8	TSN 本地 MAC 地址，只有 8 位软件可读
buf_id_used_cnt	8	Bufm 中所使用的 ID 计数器
pad0[14]	8	填充
eos_md_in_cnt	64	进入 EOS 模块的元数据计数器

eos_md_out_cnt	64	EOS 模块输出的元数据计数器
eos_q0rest_cnt	6	EOS 模块 Q0 队列的剩余长度计数器
eos_q1rest_cnt	6	EOS 模块 Q1 队列的剩余长度计数器
eos_q2rest_cnt	6	EOS 模块 Q2 队列的剩余长度计数器
eos_q3rest_cnt	6	EOS 模块 Q3 队列的剩余长度计数器
pad1	8	填充
pad2[12]	8	填充
goe_pkt_in_cnt	64	进入 GOE 模块的分组计数器
goe_port0_out_cnt	64	GOE 模块往 0 口输出的分组计数器
goe_port1_out_cnt	64	GOE 模块往 1 口输出的分组计数器
goe_discard_cnt	64	GOE 模块丢弃的分组计数器

表 5 PTP 报文结构体

struct ptp_packet		
字段	位宽	含义
dst_mac[6]	8	Mac 地址
src_mac[6]	8	端系统/交换节点 Mac 地址
eth_type	16	协议类型, 暂定为 0x88F7
transpec	4	
msg_type	4	消息类型(report:0x1F 和 update:0x2F)
ver_ptp	4	版本号
reserve0	4	保留
pkt_len	16	报文长度
domain_no	8	域号
reserve1	8	保留
flag	16	控制信息标志位
correction_field	64	透明时钟的修正域
seq	16	序列号
pad[16]	8	填充
timestamp[10]	8	时间戳
reserve2[6]	8	保留位
cfg	struct config_info	配置信息结构体
status	struct status_info	状态信息结构体

## (2) PTP 报文中写请求和写响应映射

当软件下发写请求时, PTP 中的配置字段会被重新赋值, 然后通过发送线程进行发送到交换机和 TSN 端系统设备上。交换机在收到写请求后, 会对原有的信息进行覆盖, 并返回写响应信息用于软件比较是否写入成功。目前软件需要保证写请求和写响应的正确映射, 提出写请求通信机制, 如下图所示。



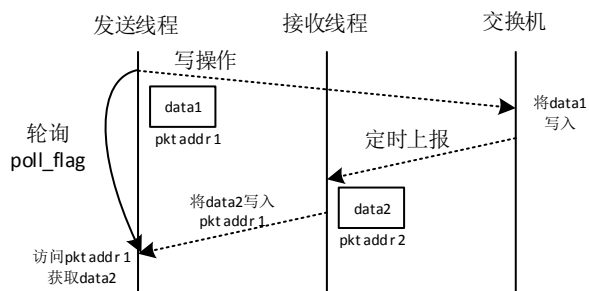


图 5 写请求通信机制

- 发送线程构造 PTP 报文，同时在报文中携带所有的控制信息 data1（不需要配置的保持原来的值不变），报文缓冲区地址为 pkt addr 1。设置轮询标志位 poll\_flag 为 0。记录目的交换机地址和写请求缓冲区地址 pkt addr 1 的映射关系。然后发送报文到交换机。
- 交换机根据 data1 对应的位置，把 data1 写入到相应的位置（交换机会把所有的配置信息都写入相应的地址中），然后读取这些地址中的值 data2 分别放在 PTP 的配置信息相应的位置，定时上报给接收线程。
- 接收线程接收到交换机发送来的 PTP 报文，报文地址为 pkt addr 2。判断 PTP 报文中的写响应标志位是否为 1，如果为 1，则说明该报文包含写响应数据。提取 PTP 报文中源交换机地址，查找映射表找到对应的写请求缓冲区地址 pkt addr 1。将写响应数据 data2 写入 pkt addr 1 中，并设置轮询标志位 poll\_flag 置为 1；
- 发送线程一直轮询标志位，当轮询标志位为 1 时，从 pkt addr 1 中获取数据，然后提取相应的配置信息进行比较。

综上所述，写通信机制中需要维护交换机地址和写请求缓冲区的地址映射关系。

表 6 交换机地址和写请求缓冲区地址映射结构体

struct addr_map		
字段	位宽	含义
sw_mac[6]	8	交换机的 mac
pkt_buf_addr	32	写请求缓冲区地址

### 3.1.3 文本格式

该文本格式是 debug 信息写入文本，格式为每个关键字对应一个值，并且关键字与关键字之间需要换行，每一组元素包含在大括号里面，写入文本的 debug 信息主要包含时间戳，序列号，交换机的 sw\_id，配置信息和状态信息

表 7 存入文本的 debug 信息格式

{
time: 2019-05-16-19:46:20
seq: 1
sw_id: 2
direction: 1
flag: 1
host_mac: 12:34:45:56:56:78
}



```
tb_rate: 10
esw_pkt_in_cnt: 20
esw_pkt_out_cnt: 30
local_mac_addr: 78: 56:45:56:56:78
buf_id_used_cnt: 40
time_slot_flag: 1
eos_q0rest_cnt: 50
eos_q1rest_cnt: 30
eos_q2rest_cnt: 20
eos_q3rest_cnt: 10
res: 0
eos_md_in_cnt: 10
eos_md_out_cnt: 20
goe_pkt_in_cnt: 30
goe_port0_out_cnt: 40
goe_port1_out_cnt: 50
goe_discard_cnt: 60
}
```

### 3.1.4 编程接口

#### (1) PTP 报文构造

函数定义	static struct fast_packet *build_fast_ptp_packet(struct ptp_packet *ptp)
输入参数	PTP 报文的结构体
返回结果	FAST 报文
功能描述	构造在 PTP 报文中携带配置信息字段的 FAST 报文，主要是用于在 PTP 报文中改变配置信息字段的值对交换机和端系统进行配置

#### (2) FAST 报文接收回调处理（作为 Fast lib 中接收回调函数）

函数定义	static void tsn_fast_packet_process(struct fast_packet *pkt, int pkt_len)
输入参数	FAST 报文,报文的长度
返回结果	无
功能描述	对接收的 FAST 报文进行解析，目前只解析 PTP 报文，根据报文中控制字段的标志位来判断是否是写响应的报文。如果标志位没有置位，则只是时间同步报文，否则该 PTP 报文中携带写响应。

#### (3) 使用写请求配置传播方向

函数定义	int ptp_write_direction(u8 *sw_mac, u8 direction)
输入参数	交换机的 MAC 地址，传播方向
返回结果	是否写成功
功能描述	修改 PTP 报文中的配置信息的 direction 字段，用于修改报文的传输

	方向，即从 0 或 1 口发送
--	-----------------

(4) 使用写请求配置端系统的 MAC 地址

函数定义	int ptp_write_host_mac(u8 *sw_mac, u8 *host_mac)
输入参数	交换机的 MAC 地址，主机的 MAC 地址
返回结果	是否写成功
功能描述	修改 PTP 报文中的配置信息的 host_mac 字段,用于修改主机的 MAC 地址

(5) 使用写请求配置令牌桶参数 rate

函数定义	int ptp_write_tb_rate(u8 *sw_mac, u32 rate)
输入参数	交换机的 MAC 地址，令牌桶产生令牌的速率 rate
返回结果	是否写成功
功能描述	修改 PTP 报文中的配置信息的 tb_rate 字段，用于配置令牌桶产生令牌的速率。

(6) 对 PTP 中携带的配置信息处理

函数定义	static void ptp_write_reply_process(struct ptp_packet *pkt)
输入参数	ptp 报文
返回结果	无
功能描述	在 PTP 报文中提取出配置信息字段，然后进行分发，映射到相应到缓冲区中，然后进行比较，是否与写的值相同。

(7) 对 PTP 中携带的配置信息和状态信息写入到文件

函数定义	static void ptp_debug_info_process(struct ptp_packet *pkt)
输入参数	ptp 报文
返回结果	无
功能描述	根据 PTP 中携带的 debug 信息，写入到相应的文件。

### 3.1.5 处理流程

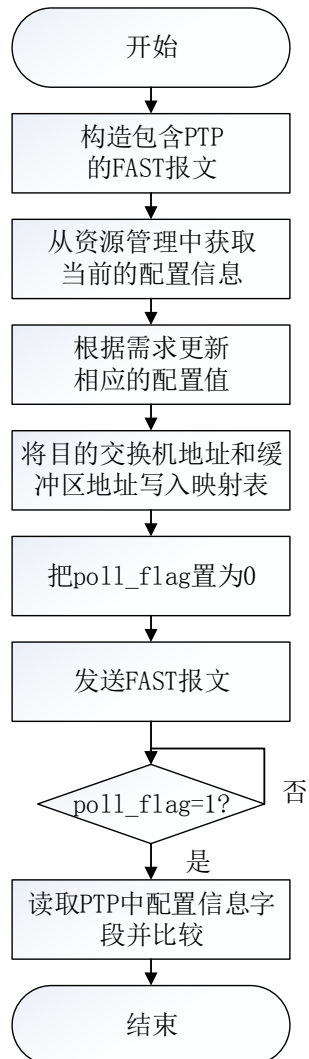


图 6 FAST 报文接收处理

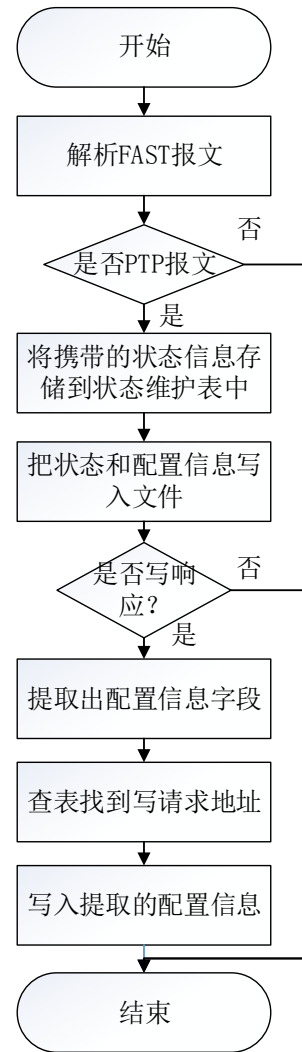


图 7 写请求流程

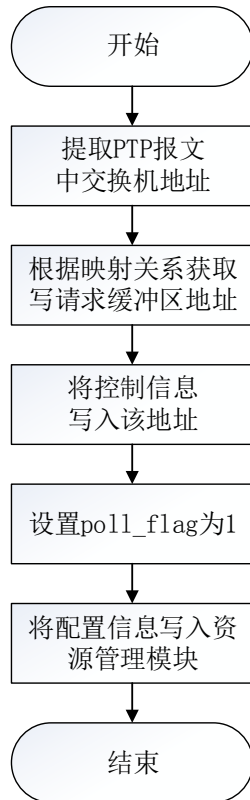


图 8 写响应处理

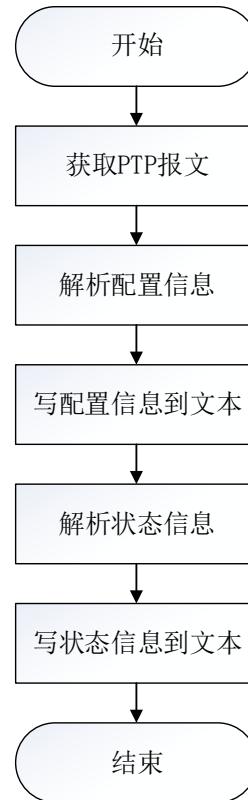


图 9 把 debug 信息写入文本

## 3.2 北向 I/O 通信库

### 3.2.1 功能描述

北向 I/O 通信库主要用于与远端的 WEB 服务器建立连接，通过主动方式上报可视化信息。其主要包括北向协议报文模块和报文 I/O 操作模块。

- (1) **协议报文模块：**主要包括对时钟同步状态报文，链路故障报文的定义。
- (2) **报文 I/O 操作模块：**主要包括构造北向协议报文，IO 初始化，发送北向报文。

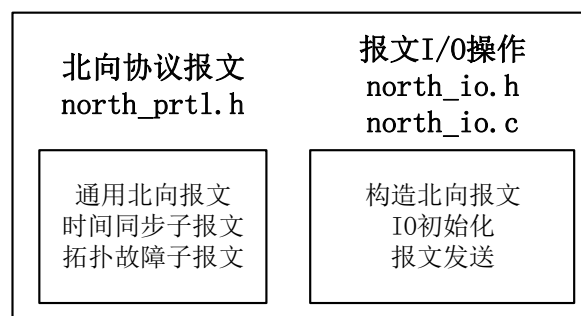


图 10 北向 I/O 通信模块

### 3.2.2 数据结构

- (1) 通用北向报文

表 8 通用北向报文结构体

struct north_packet		
字段	位宽	含义
version	8	版本号
type	8	报文类型：时钟同步报文，拓扑报文
len	16	整个报文长度
data[0]	8	北向报文数据

表 9 北向报文类型

enum NORTH_PKT_TYPE		
类型	数值	含义
CLOCK_SYNC	0	时钟同步报文
LINK_ERR	1	链路故障报文

## (2) 时钟同步状态子报文

表 10 时钟同步状态 JSON 格式

<pre>{   "num": "6",   "sync state":[     {       "id": "7";//交换机或主机 ID       "type": "host",//交换机或主机       "value": "0x889898"//同步状态值     },     {       "id": "8";       "type": "switch",       "value": "0x990234"     }   ] }</pre>
--

## (3) 链路故障子报文

表 11 链路故障 JSON 格式

<pre>{   "src switch": "1",//源交换机 ID   "dst switch": "2"//目的交换机 ID }</pre>
--

## 3.2.3 编程接口

### (1) 北向报文构造

函数定义	struct north_packet* build_north_packet(u8 type, u8 *data, u32 data_len)
------	--

输入参数	报文类型，数据起始地址，数据长度
返回结果	北向报文
功能描述	构造北向报文，并填充数据

### (2) 北向通信初始化

函数定义	int init_north_io()
输入参数	无
返回结果	初始化北向 IO 库
功能描述	北向通信初始化

### (3) 北向报文发送

函数定义	int send_north_packet(u8 *pkt, int len)
输入参数	报文地址，报文长度
返回结果	是否发送成功
功能描述	向北向应用发送北向报文

## 3.3 拓扑管理模块

### 3.3.1 功能描述

拓扑管理模块主要对整个 TSN 网络中交换机与交换机，交换机和主机之间连接关系进行管理。资源分配模块会根据拓扑信息为流特征计算处理的路径。

- (1) **拓扑结构**：拓扑结构包括对交换机和端系统属性以及连接关系定义。
- (2) **拓扑管理**：块主要包括通过文本对拓扑初始化，节点的插入，主机 MAC 地址修改，获取节点属性等功能。

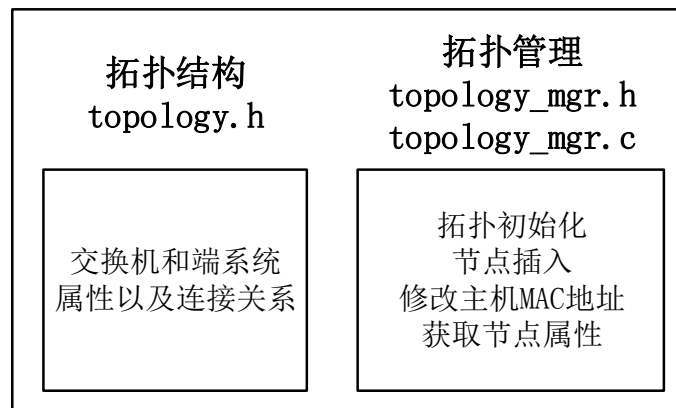


图 11 拓扑管理模块

### 3.3.2 数据结构

拓扑管理中既需要保存交换机之间连接关系，同时需要保存交换机和主机之间连接关系。采用双向链表的形式，链表的成员需要保存交换机和主机的 ID 和 MAC 地址，用于保存交换机的信息，并记录与上一跳和下一跳连接的端口。目前默认的连接关系为上一跳交换机 1 端口连接下一跳交换机的 0 端口。

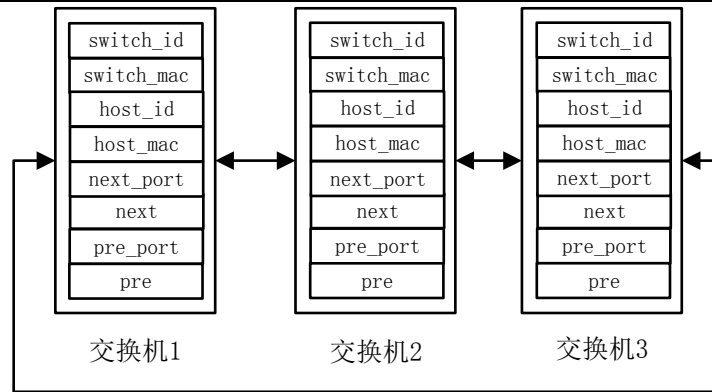


图 12 拓扑数据结构

(1) 交换机节点

表 12 交换机节点信息

struct switch		
字段	位宽	含义
sw_id	8	交换机 ID
host_id	8	主机 ID
sw_mac[6]	8	交换机 MAC
host_mac[6]	8	主机 MAC
next_port	8	与下一跳交换机连接的端口编号
prev_port	8	与上一跳交换机连接的端口编号
next	struct switch*	下一跳交换机地址
prev	struct switch*	上一跳交换机地址

表 13 拓扑结构体

struct topology		
字段	位宽	含义
head	struct switch *	头结点
tail	struct switch *	尾节点
size	32	交换机的个数

### 3.3.3 文本格式

从拓扑描述文件中读取文件内容，并在程序中把这些内容以链表的方式保存起来。每个关键字对应一个值，并且关键字与关键字之间需要换行，每一组信息中使用大括号进行隔开，每组信息主要包含交换机的 id 和 mac，端系统的 id 和 mac，还有连接的端口号。

表 14 拓扑文件格式

```
{
  sw_id: 1
  sw_mac: 12:34:45:56:56:78
  host_id: 1
```



```
host_mac: 78: 56:45:56:56:78
next_port: 1
pre_port: 0
}
{
  sw_id: 2
  sw_mac: 12:34:45:56:56:78
  host_id: 2
  host_mac: 78: 56:45:56:56:78
  next_port: 1
  pre_port: 0
}
```

### 3.3.4 编程接口

#### (1) 初始化交换机链表

函数定义	int init_topology()
输入参数	无
返回结果	初始化是否成功
功能描述	从文本读取交换机和主机的信息，使用双向链表的形式存放起来

#### (2) 修改交换机中连接主机的 MAC 地址

函数定义	int modify_host_mac_with_sw (u8 sw_id, u8 *mac)
输入参数	交换机 ID, 主机 mac 地址
返回结果	返回 0 修改成功，否则修改失败
功能描述	根据 sw_id 查找到需要修改的交换机，然后把传递过来的参数替换原来结构体中的值

#### (3) 插入交换机节点

函数定义	int insert_node_to_topology(struct switch *sw)
输入参数	交换机节点信息
返回结果	是否插入成功
功能描述	将新的交换机节点插入到双向链表中

#### (4) 获取交换机的 MAC 地址

函数定义	u8 *get_sw_mac_by_id(u8 sw_id)
输入参数	交换机的 sw_id
返回结果	交换机的 MAC 地址指针
功能描述	根据交换机的 sw_id, 找到对用的节点，然后获取交换机的 MAC 地址

#### (5) 根据交换机机的 MAC 获取交换机 ID

函数定义	u8 get_sw_id_with_mac (u8 *sw_mac)
------	------------------------------------

输入参数	交换机 MAC 地址
返回结果	交换机 ID
功能描述	根据交换机 MAC 地址获取该交换机 ID

(6) 根据主机 id 获取交换机的 id

函数定义	u8 get_sw_id_by_host_id(u8 host_id)
输入参数	主机的 host_id
返回结果	交换机的 sw_id
功能描述	根据主机的 host_id 来找到对应的节点，获取该节点的 sw_id

(7) 根据交换机的 ID 获取主机的 MAC 地址

函数定义	u8* get_host_mac_by_sw_id(u8 sw_id)
输入参数	交换机的 host_id
返回结果	主机的 MAC 地址
功能描述	根据交换机的 sw_id 来找到对应的节点，获取该节点的 host_mac

(8) 配置交换机上的主机 MAC 地址

函数定义	int config_host_mac(u8 sw_id, u8 *host_mac)
输入参数	交换机的 id, 需要配置的主机的 mac 地址
返回结果	配置成功和失败
功能描述	根据交换机的 sw_id 获取需要配置的交换机，然后改变该节点的 host_mac 地址，更新拓扑，并配置到硬件中。

### 3.3.5 处理流程

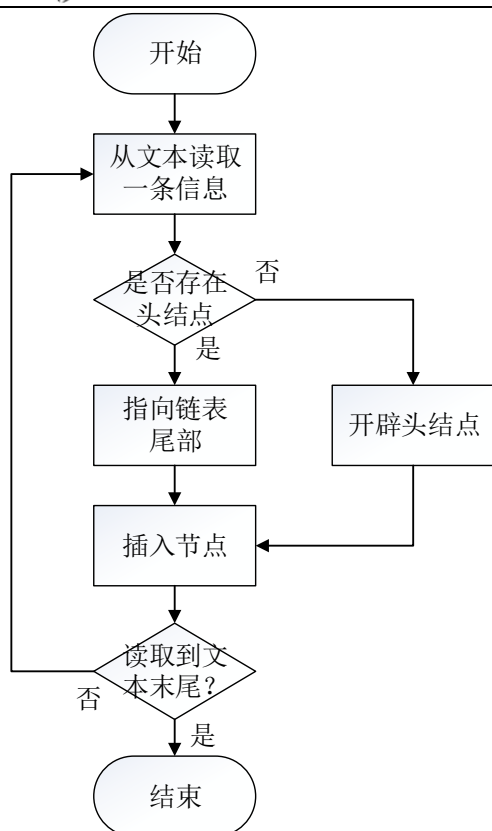


图 13 拓扑初始化

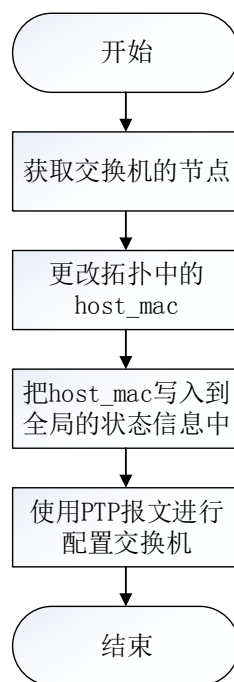


图 14 配置交换机的 host\_mac 地址

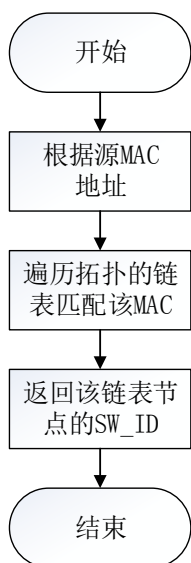


图 15 源 MAC 查找到 SW\_ID

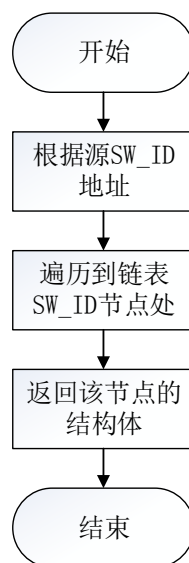


图 16 获取交换机的信息

### 3.4 全局资源管理模块

#### 3.4.1 功能描述

全局资源管理是对所有 TSN 交换机的可调度资源的管理，主要包括设备中带宽、CQF 队列资源、控制信息。资源调度分配模块会根据设备资源的状况进行调度决策。

- (1) **设备资源**：包括带宽资源，CQF 资源和当前配置信息。
  - a) **带宽资源**：带宽资源主要用于带宽预约型流量调度使用，但是需要考虑 TSN 流量也会占用带宽资源。
  - b) **CQF 资源**：用于时间敏感型流量，标识 CQF 队列的使用情况。
  - c) **配置信息**：用来保存当前设备的配置信息。主要有方向 `direction`，令牌桶的速率 `tb_rate`，还有主机的 `mac` 地址。
- (2) **资源管理**：主要包括资源初始化，获取设备资源，更新设备资源，更新配置信息。
  - a) **资源初始化**：在开始时对每一个设备的资源进行初始化，主要是对总带宽和总的 CQF 队列长度进行赋值。
  - b) **获取设备资源**：获取设备当前的资源的使用情况。
  - c) **更新设备资源**：提供更新设备资源的接口，可以通过带宽或者 CQF 改变量来更新当前设备的资源状态。
  - d) **更新配置信息**：提供更新配置信息的接口，可以根据配置信息的改变来更新当前的配置信息。

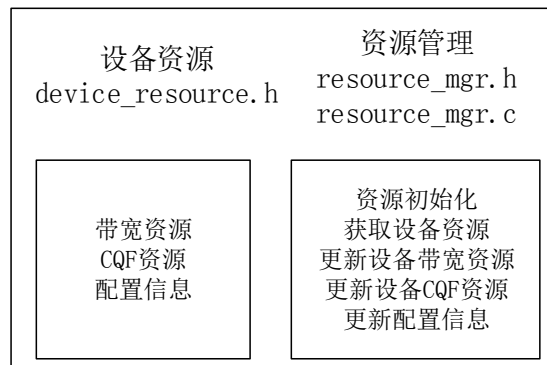


图 17 资源管理模块

### 3.4.2 数据结构

每个 TSN 交换机中有 4 个队列，其中前两个个队列为 CQF 队列，用于存储 TSN 流；第 3 个队列为带宽预约队列，用于存储带宽预约流；第 4 个队列为 Best-effort 队列，用于存储 Best-effort 流量。可调度的资源包括带宽资源和 CQF 资源

- (1) 带宽资源包括总的带宽、已用带宽、空闲带宽和令牌桶速率。其中已用带宽指 TSN 流和带宽预约流总共占用的带宽资源。令牌桶速率为带宽预约流占用的带宽资源。
- (2) CQF 队列包括总共的队列长度，已用的队列长度和空闲的队列长度。其中 CQF 队列分为奇偶时间槽，流量为周期性流量，在每个时间槽上队列资源的使用情况不相同。
- (3) 配置信息中包含报文的传输方向，产生令牌的速率 `rate` 和端系统 MAC 地址。

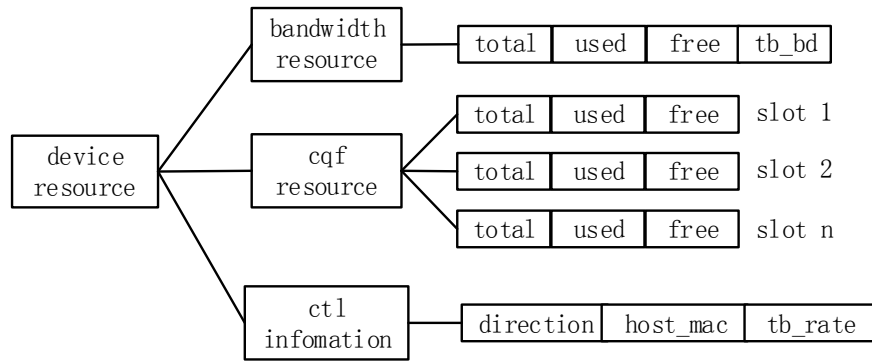


图 18 设备资源数据结构

### (1) CQF 队列资源

表 15 时间槽中 CQF 队列结构体

struct cqf_resource		
字段	位宽	含义
total_len	32	CQF 队列的总长度
free_len	32	空闲 CQF 队列的长度
used_len	32	已用 CQF 队列的长度

### (2) 带宽资源

表 16 端口的带宽结构体

struct bd_resource		
字段	位宽	含义
total_bd	32	端口的总带宽
free_bd	32	空闲的带宽
used_bd	32	端口已经使用的带宽
tb_bd	32	令牌桶已用带宽

### (3) 设备资源

表 17 TSN 资源管理结构体

struct device_res		
字段	位宽	含义
sw_id	8	交换机 sw_id
bd_res	struct bd_resource	带宽的结构体
cqf_res[SLOT_NUM]	struct cqf_resource	CQF 队列的结构体
cfg	struct config_info	配置信息结构体 (南向接口中定义)

### (4) 全局资源

表 18 TSN 资源管理结构体

struct global_resource		
字段	位宽	含义

cur_num	32	当前设备的数量
device[MAX_NODE_NUM]	struct device_resource	全局设备资源

#### (5) 全局资源描述

全局资源中 CQF 队列长度和带宽资源通过宏定义设置。节点数目和配置信息中 host MAC 可以通过拓扑信息进行获取。

表 19 全局资源宏定义

#define NODE_NUM 7
#define CQF_QUEUE_LEN 1000
#define TOTAL_BD 1000

### 3.4.3 编程接口

#### (1) 初始化设备资源

函数定义	int init_global_resource()
输入参数	无
返回结果	0 代表初始化成功，1 代表初始化失败
功能描述	初始化 struct global_resource 结构体，初始化时根据宏定义进行初始化，host_mac 则需要根据拓扑模块填充对应的 MAC 地址，两个端口共用相同的队列，每个设备的资源的结构体又包含 cqf_res、bd_res 和 ctl 结构体，需要分别进行填充。

#### (2) 获取交换机的剩余带宽资源的状态

函数定义	u32 get_free_bd_resource(u8 sw_id)
输入参数	交换机 ID
返回结果	剩余带宽
功能描述	根据 sw_id 来获取当前该交换机剩余带宽的资源状态

#### (3) 获取交换机的令牌桶带宽资源的状态

函数定义	u32 get_tb_bd_resource(u8 sw_id)
输入参数	交换机 ID
返回结果	已用的令牌桶带宽
功能描述	根据 sw_id 来获取当前该交换机已用的令牌桶带宽的资源状态

#### (4) 获取交换机的剩余 CQF 队列的资源状态

函数定义	u32 get_free_cqf_resource(u8 sw_id)
输入参数	交换机 ID
返回结果	剩余 CQF 队列
功能描述	根据 sw_id 来获取当前该交换机剩余的 CQF 队列的资源状态

#### (5) 更新交换机的带宽

函数定义	int update_bd_resource(u8 sw_id, u32 bd_alloc)
输入参数	需要更新的交换机 sw_id，带宽改变的值
返回结果	更新成功返回 0，失败返回-1

功能描述	更新 SW_ID 代表的交换机带宽的状态
------	----------------------

(6) 更新交换机的 CQF 队列资源

函数定义	int update_cqf_resource(u8 sw_id, u32 queue_alloc)
输入参数	交换机 ID, 改变的队列长度
返回结果	更新成功返回 0, 失败返回-1
功能描述	更新 SW_ID 代表的交换机 CQF 队列的状态

(7) 更新交换机当前的配置信息

函数定义	int update_ctrl_info(u8 sw_id, struct config_info cfg)
输入参数	交换机 ID, 配置信息的结构体
返回结果	更新成功返回 0, 失败返回-1
功能描述	通过交换机 ID 来更新当前交换机的配置信息

(8) 全局资源描述文件

3.4.4 处理流程

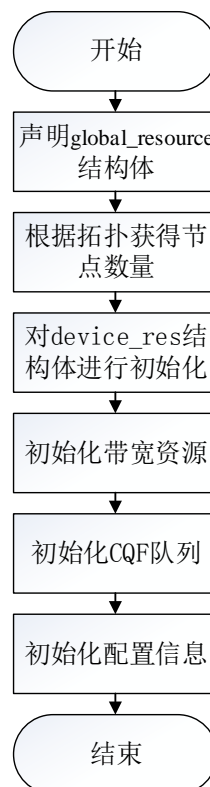


图 19 全局资源初始化



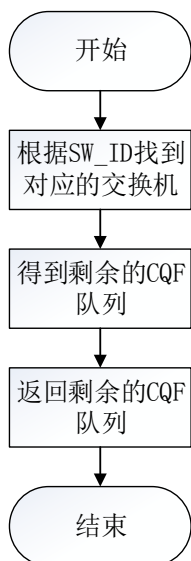


图 20 获取设备 CQF 队列资源状态

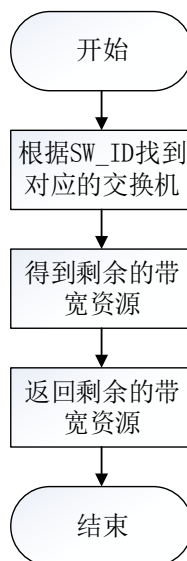


图 21 获取设备带宽资源状态

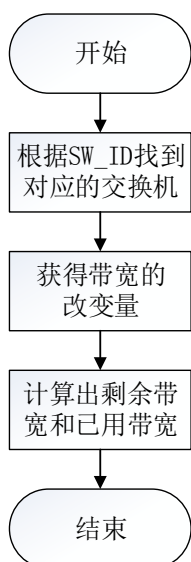


图 22 更新带宽的占用情况

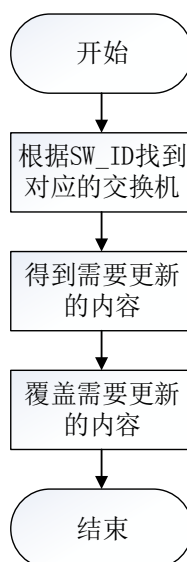


图 23 更新配置信息

## 3.5 资源预约模块

### 3.5.1 功能描述

资源预约模块主要功能是读取资源预约流的特征信息进行保存，其主要包括流特征和预约管理模块。

- (1) **流特征模块**：主要描述 TSN 单播流和带宽约束单播流的特征。
- (2) **预约管理模块**：主要通过读文本初始化流资源预约表。

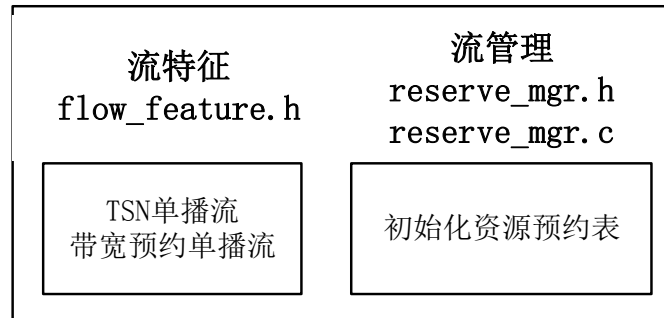


图 24 资源预约模块

### 3.5.2 数据结构

资源预约表中主要包括 TSN 单播流和带宽预约单播流两种。

**TSN 单播表项：**主要包括源端系统 MAC、目的端系统 MAC、优先级、发送周期、报文数量、报文大小和最大延迟。TSN 单播流的优先级为 6 或 7。报文在网络上的整个传输时间需要小于最大延迟。

**带宽预约单播表项：**主要包括源端系统 MAC、目的端系统 MAC，优先级和带宽。带宽预约单播流的优先级为 4 或 5。

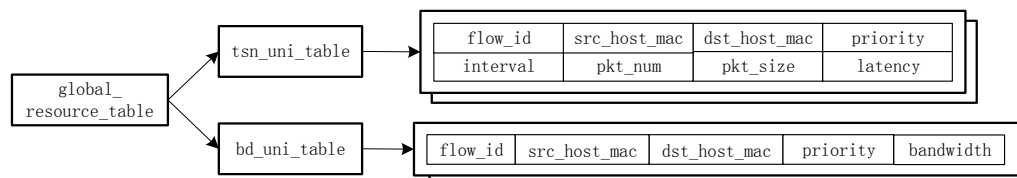


图 25 资源预约表

#### (1) TSN 单播流特征

表 20 TSN 单播流流特征结构体

struct tsn_uni_feature		
字段	位宽	含义
flow_id	16	流 ID
src_host_mac[6]	8	源端系统 MAC 地址
dst_dst_mac[6]	8	目的端系统 MAC 地址
Priority	8	优先级 (6,7)
Interval	8	时间周期
pkt_num	16	周期内发送数量
pkt_size	16	报文大小
latency	16	最大传输延迟

#### (2) TSN 单播流表

表 21 TSN 单播流表结构体

struct tsn_uni_table		
字段	位宽	含义
cur_num	u8	当前表项数目
tsn_uni[MAX_FLOW_NUM]	struct tsn_uni_feature	流特征

## (3) 带宽预约单播流特征

表 22 带宽预约单播流特征结构体

struct bd_uni_feature		
字段	位宽	含义
flow_id	32	流 ID
src_host_mac[6]	8	源端系统 MAC 地址
dst_host_mac[6]	8	目的端系统 MAC 地址
Priority	8	优先级 (4,5)
bandwidth	16	带宽约束

## (4) 带宽预约单播流表

表 23 带宽预约单播流表结构体

struct bd_uni_table		
字段	位宽	含义
cur_num	8	当前表项数目
bd_uni[MAX_FLOW_NUM]	struct bd_uni_feature	流特征

## (5) 资源预约表

表 24 资源预约表结构体

struct resource_reserve_table		
字段	位宽	含义
tsn_uni_tbl	struct tsn_uni_table	TSN 单播流表
bd_uni_tbl	struct bd_uni_table	带宽预约单播流表

## 3.5.3 文件格式

从资源预约表文件中读取文件内容并保存起来。每个关键字对应一个值，并且关键字与关键字之间需要换行，每一组信息中使用大括号进行隔开。每组信息主要包含流类型，流的特征等。

表 25 资源预约表文件格式

{
type: tsn
flow_id: 1
src_mac: 12:34:45:56:56:78
dst_mac: 78:56:45:56:56:78
priority: 6
interval: 125
pkt_num: 3
pkt_size: 1500
latency: 1000
}
{
type: bd

```

flow_id: 1
src_mac: 12:34:45:56:56:78
dst_mac: 78:56:45:56:56:78
priority: 4
bandwidth: 125
}

```

### 3.5.4 编程接口

#### (1) 初始化资源预约表

函数定义	struct global_resource_table *init_global_resource_table ()
输入参数	用户资源预约文件
返回结果	初始化是否成功
功能描述	初始化资源预约表

### 3.5.5 处理流程

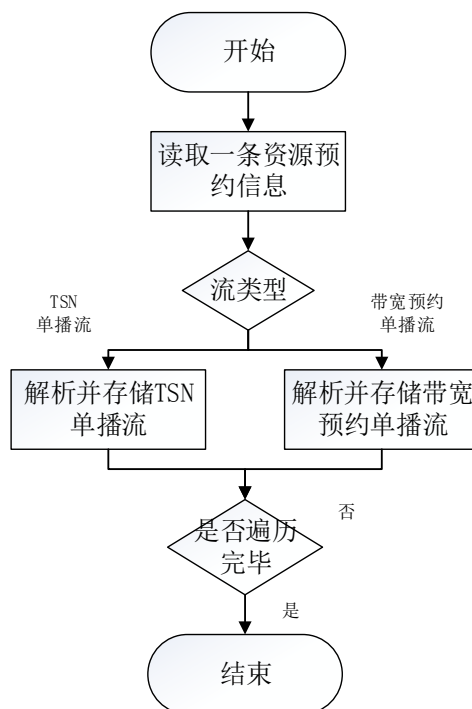


图 26 初始化资源预约表

## 3.6 资源调度分配模块

### 3.6.1 功能描述

资源调度模块的主要功能是根据流资源预约表信息进行流量的调度和资源分配。其主要包括路径计算，流量排序，时隙调度，带宽调度和资源配置五个模块。

- (1) **路径计算模块：**根据拓扑和流特征信息计算每条流经过的交换机路径。
- (2) **流量排序模块：**根据路径、报文数量、带宽等条件对流量进行排序。
- (3) **时隙调度模块：**根据每个时间槽可用的 CQF 队列长度对 TSN 单播流进行调度。
- (4) **带宽调度模块：**根据可用的链路带宽对带宽预约流进行调度。

- (5) **资源配置模块**：包括两方面功能：(1) 根据资源调度分配结果对交换机和端系统进行配置；(2) 对全局资源数据进行更新。

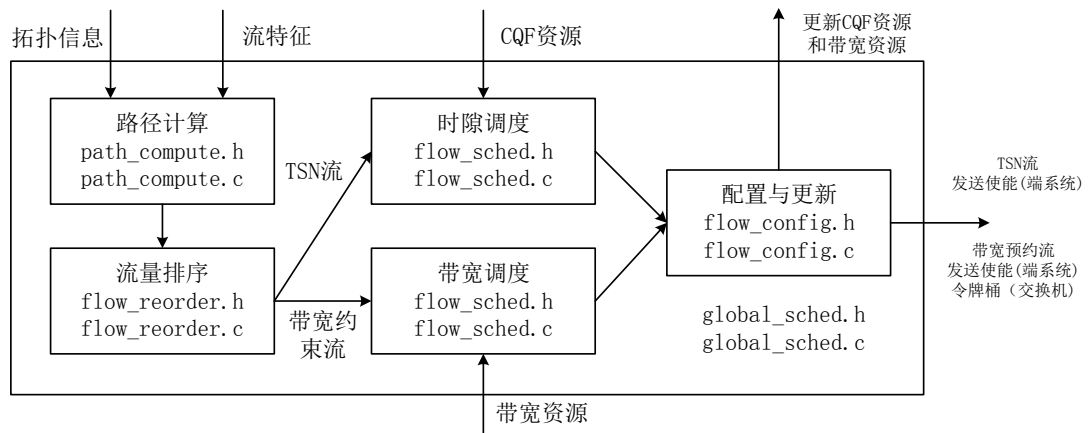


图 27 资源调度分配模块

### 3.6.2 数据结构

- (1) 单播路径信息

表 26 单播路径信息结构体

struct uni_path		
字段	位宽	含义
flow_id	16	流 ID
path_len	8	路径长度
src_sw_id	8	源交换机 ID
dst_sw_id	8	目的交换机 ID

- (2) TSN 单播流调度信息

表 27 TSN 单播流调度信息结构体

struct tsn_sched_info		
字段	位宽	含义
path	struct uni_path	单播路径
interval	8	周期
pkt_num	16	报文数量
pkt_size	16	报文大小
latency	16	延迟

- (3) TSN 单播流调度信息集合

表 28 TSN 单播流调度信息集合结构体

struct tsn_sched_set		
字段	位宽	含义
cur_num	u32	当前数量
tsn[MAX_FLOW_NUM]	struct tsn_sched_info	所有 TSN 流调度信息

- (4) 带宽预约单播流调度信息

表 29 带宽预约单播流调度信息结构体

struct bd_sched_info		
字段	位宽	含义
path	struct uni_path	单播路径
bandwidth	16	带宽

## (5) 带宽预约单播流调度信息集合

表 30 带宽预约单播流调度信息集合结构体

struct bd_sched_set		
字段	位宽	含义
cur_num	u32	当前数量
bd[MAX_FLOW_NUM]	struct bd_sched_info	所有带宽预约流调度信息

## (3) TSN 单播流配置信息

表 31 TSN 单播流配置信息结构体

struct tsn_uni_cfg		
字段	位宽	含义
tx_enable	8	发送使能
host_mac[6]	8	源端系统 MAC 地址

## (4) 带宽预约单播流配置信息

表 32 带宽预约单播流配置信息结构体

struct bd_uni_cfg		
字段	位宽	含义
tx_enable	8	发送使能
host_mac[6]	8	源端系统 MAC 地址
path_len	8	路径长度
sw_mac[MAX_NODE_NUM][6]	8	所有交换机 MAC 地址
tb_rate [MAX_NODE_NUM]	16	所有交换机中令牌桶带宽

## 3.6.3 编程接口

## (1) 全局流资源调度

函数定义	int global_flow_resource_sched(struct global_resource_table *g_table)
输入参数	全局流资源预约表
返回结果	配置成功的流数目
功能描述	为所有流量调度和分配资源

## (2) 单播路径计算

函数定义	struct uni_path compute_uni_path(u8 *src_host_mac, u8 *dst_host_mac)
输入参数	端系统源 MAC 和目的 MAC
返回结果	单播路径信息
功能描述	计算单播路径信息

## (3) 计算所有 TSN 单播流调度信息

函数定义	struct ts_n_sched_set compute_ts_n_uni_path(struct ts_n_uni_table *ts_n_tbl)
输入参数	TSN 单播流表
返回结果	所有 TSN 单播流调度信息集合
功能描述	为每条 TSN 单播流计算路径信息，并生成流调度信息集合

## (4) 计算所有带宽预约单播流调度信息

函数定义	struct bd_sched_set compute_bd_uni_path(struct ts_n_bd_table *bd_tbl)
输入参数	带宽预约单播流表
返回结果	所有带宽预约单播流调度信息集合
功能描述	为每条带宽预约单播流计算路径信息，并生成流调度信息集合

## (5) 对所有 TSN 单播流排序

函数定义	struct ts_n_sched_set ts_n_flow_reorder_by_pkt_num (struct ts_n_sched_set ts_n_set)
输入参数	所有 TSN 单播流调度信息集合
返回结果	排序后的 TSN 单播流调度信息集合
功能描述	根据报文数量对所有 TSN 单播流排序

## (6) 对所有带宽预约单播流排序

函数定义	struct bd_sched_set bd_flow_reorder_by_bd (struct bd_sched_set bd_set)
输入参数	所有带宽预约单播流调度信息集合
返回结果	排序后的带宽预约单播流调度信息集合
功能描述	根据带宽大小对所有 TSN 单播流排序

## (7) TSN 单播流时隙调度计算

函数定义	struct ts_n_uni_cfg ts_n_uni_sched(struct ts_n_sched_info ts_n_info)
输入参数	TSN 单播流调度信息
返回结果	配置信息
功能描述	TSN 单播流时隙调度

## (8) 带宽预约单播流带宽调度计算

函数定义	struct bd_uni_cfg bd_uni_sched (struct bd_sched_info bd_info)
输入参数	带宽预约单播流调度信息
返回结果	配置信息
功能描述	带宽预约单播流时隙调度

## (9) TSN 单播流配置

函数定义	int config_ts_n_uni_flow(struct ts_n_uni_cfg ts_n_cfg)
输入参数	TSN 单播流配置信息
返回结果	是否配置成功
功能描述	TSN 单播流配置



## (10) 带宽预约单播流配置

函数定义	int config_bd_uni_cfg (struct bd_uni_cfg bd_cfg)
输入参数	带宽预约单播流配置信息
返回结果	是否配置成功
功能描述	带宽预约单播流配置

## (11) 为 TSN 单播流更新资源

函数定义	int upate_tsn_uni_resource (struct tsn_sched_info tsn_info)
输入参数	TSN 单播流调度信息
返回结果	是否更新成功
功能描述	更新 TSN 单播流对应的带宽和 CQF 资源

## (12) 为带宽预约单播流更新资源

函数定义	int upate_bd_uni_resource (struct bd_sched_info bd_info)
输入参数	带宽预约单播流调度信息
返回结果	是否更新成功
功能描述	更新带宽预约单播流对应的带宽资源

## 3.6.4 处理流程

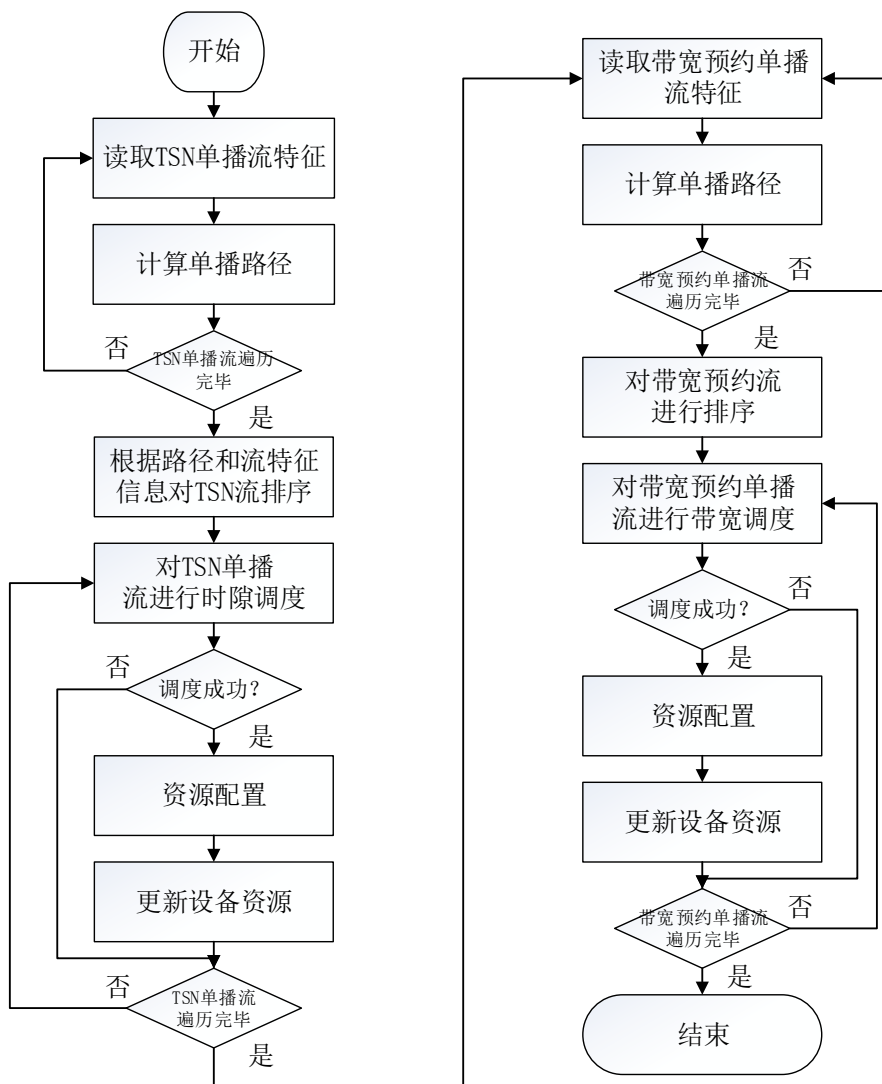


图 28 流资源调度

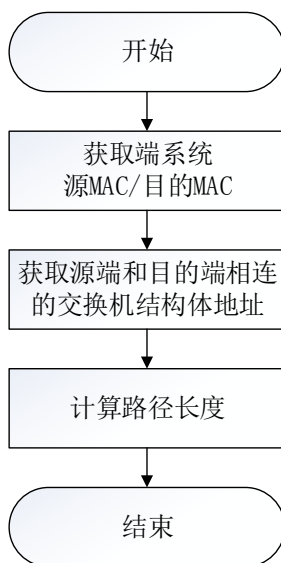


图 29 单播流路径计算

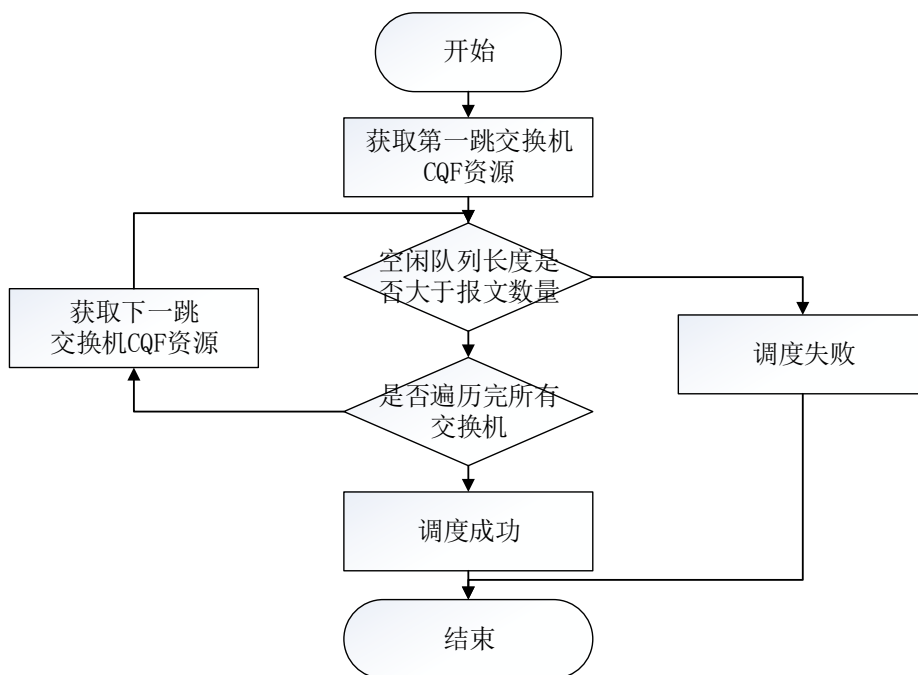


图 30 TSN 流时隙调度

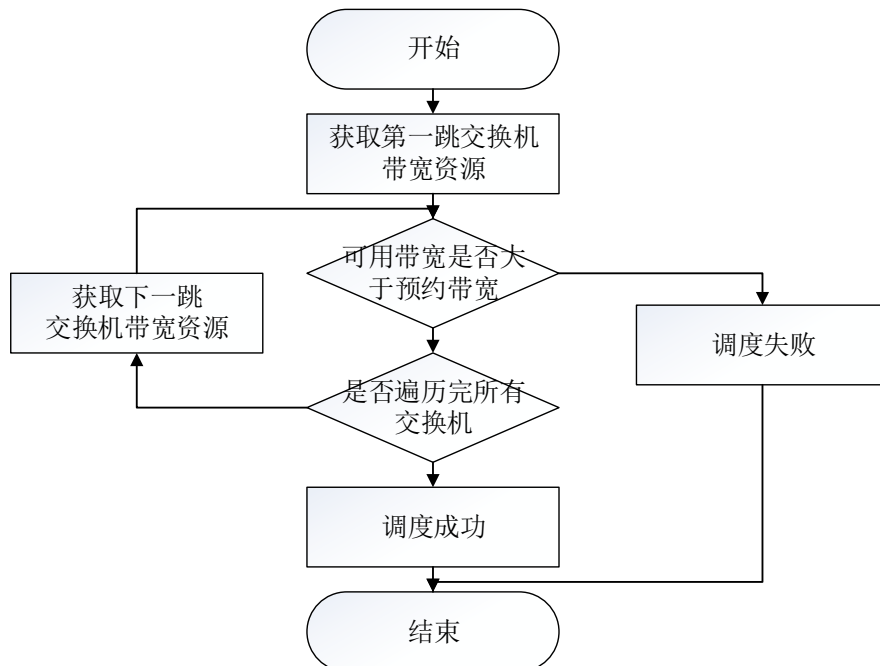


图 31 带宽预约流带宽调度

## 3.7 状态监测模块

### 3.7.1 功能描述

状态监测主要功能是维护 PTP 上报的节点的时钟同步信息，以及各个节点的状态。其主要包括状态表和状态管理模块组成。

- (1) **状态表：**通过单链表维护每一轮主机和交换机上报的 PTP 报文中时钟同步信息和节点状态。

(2) **状态管理模块：**主要提供 PTP 报文处理和状态表遍历功能。

- a) **PTP 报文处理：**PTP 报文中包含每个 TSN 交换机和 TSN 端系统传输报文累积的透明时钟。包含该 PTP 报文的 FAST 报文头部包含接收时间戳。通过将接收时间戳减去累计的透明时钟可以得到每个 TSN 节点报文的发送时间。此外，如果收到节点上报的 PTP 报文，则说明该节点的状态为运行状态。
- b) **状态表遍历功能：**定时对状态表中每个节点进行遍历，每一轮的 PTP 报文接收超时，则将该轮同步结果上报给控制器。如果超时，没有收到 TSN 交换节点发送的 PTP 报文，则说明有链路故障，进行故障上报，并进行故障恢复。目前本项目只针对单条链路故障检测。如果链路发生故障，则控制器不能接收到该条链路后续节点发送的 PTP 报文，从而可以定位故障链路。

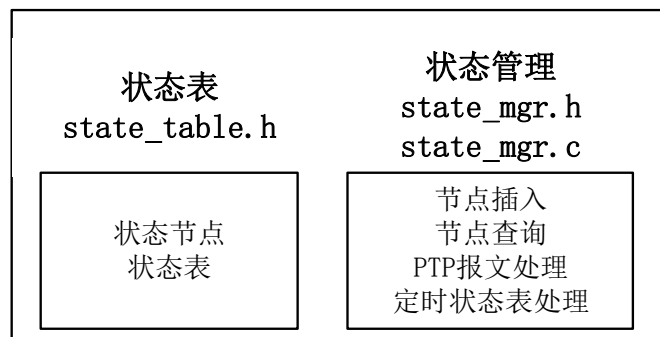


图 32 状态监测

### 3.7.2 数据结构

在 TSN 网络中每个 TSN 交换节点和 TSN 端系统会定时上报 PTP 报文。由于传输路径长度不同，可能出现还未收到第一轮发送的所有 PTP 报文，第二轮的部分 PTP 报文已经收到。因此需要为每一轮的 PTP 报文创建一个状态节点，每个节点上的 PTP 报文的序列号相同。状态监测节点中包括的信息主要包括 PTP 序列号，节点的创建时间，端系统的状态和交换机状态。

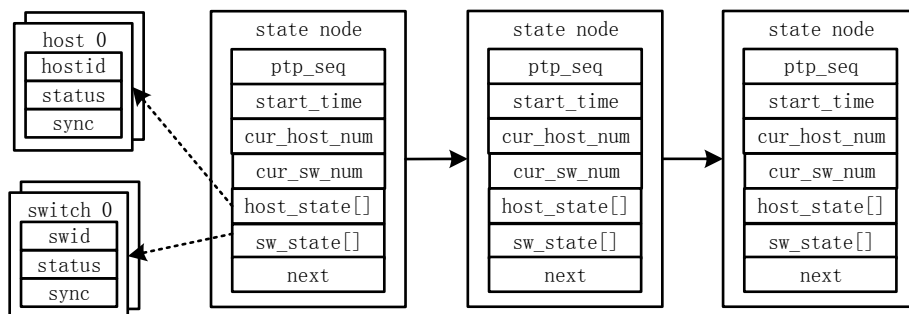


图 33 状态表结构

(1) 状态信息

表 33 状态信息结构体

struct state_info		
字段	位宽	含义
id	8	TSN 交换机/主机 ID

state	8	节点状态
sync	64	同步状态数据

## (2) 状态节点

表 34 状态节点结构体

struct state_node		
字段	位宽	含义
seq	8	PTP 报文序列号
create_time	32	节点创建时间
cur_host_num	8	当前接收到 PTP 主机数量
cur_sw_num	8	当前接收到 PTP 交换机数量
host_state[MAX_HOST_NUM]	struct state_info	主机状态集合
switch_state[MAX_SW_NUM]	struct state_info	交换机状态集合
next	struct state_node*	下一个状态节点地址

## (3) 状态表

表 35 状态表结构体

struct state_list		
字段	位宽	含义
head	struct state_node*	状态表的首节点
tail	struct state_node*	状态表尾节点
size	32	状态表节点数目

### 3.7.3 编程接口

#### (1) 向状态表插入节点

函数定义	int insert_node_to_state_list(struct state_node *node)
输入参数	状态节点
返回结果	是否插入成功
功能描述	向状态链表中插入状态节点

#### (2) 根据 seq 查询状态节点

函数定义	struct state_node* lookup_state_node_with_seq(u32 seq)
输入参数	序列号
返回结果	返回相应的状态节点，如果未找到，则返回 NULL
功能描述	根据 seq 查询状态节点

#### (3) PTP 报文处理

函数定义	void ptp_packet_state_process (struct ptp_packet *pkt)
输入参数	PTP 报文
返回结果	无
功能描述	对 PTP 报文进行解析处理，将相应的信息存储到状态表中

#### (4) 状态表处理

函数定义	void state_list_process ()
输入参数	无
返回结果	无
功能描述	对状态表进行遍历处理

#### (5) 定时进行状态表处理

函数定义	void create_timer_process_state_list (u32 interval)
输入参数	时间间隔
返回结果	无
功能描述	创建线程启动定时器，定时调用状态表处理函数

#### (6) 链路故障恢复

函数定义	int link_err_recovery(u8 sw_id)
输入参数	发生故障链路的起始交换机 ID
返回结果	链路故障恢复是否成功
功能描述	通过改变发送方向对链路故障进行修复

### 3.7.4 链路故障恢复原理

本项目当前只考虑单条链路故障的恢复。当存在某条链路发生故障时，环形链路会转变为线性链路。环形链路中，只需要通过单向传输即可到达任意的目的地，而线性链路中需要进行双向传输才能到达任意的目的地。TSN 网络中流量包括控制流量（PTP 流量）和数据流量（TSN 流和带宽预约流）。在本项目中，优先保证控制流量到达控制器。当发生故障时，通过调整交换节点的传输方向保证控制流量正常达到控制器。但是某一些数据流量无法正常达到目的地。

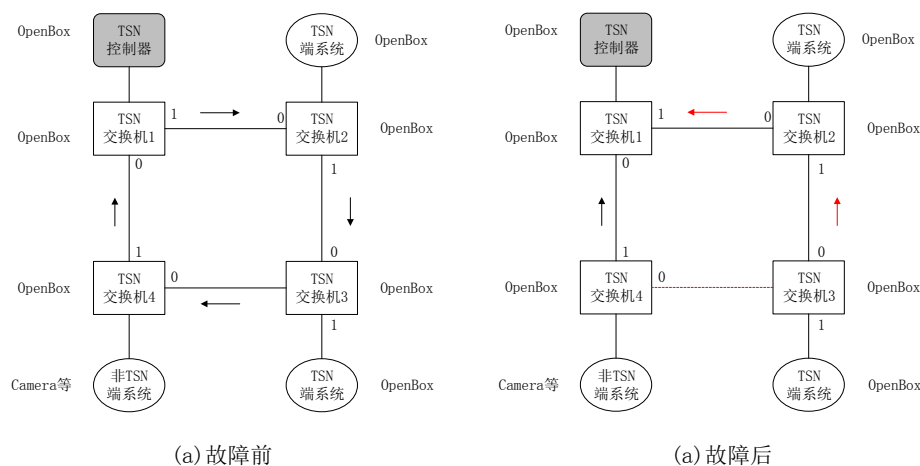


图 34 故障修复原理

在 TSN 网络中，当链路没有出现故障的时，所有经过交换机的报文都是默认从 0 进 1 出，交换机上连接的端系统是从 1 口出。

如果红色链路中发生故障，控制器会通过 PTP 报文中携带的控制信息的 direction 字段，使交换机 2,3 产生的报文从 0 口出，TSN 交换机 4 传输方向不变。这样可以保证控制报文顺利到达控制器。

## 3.7.5 处理流程

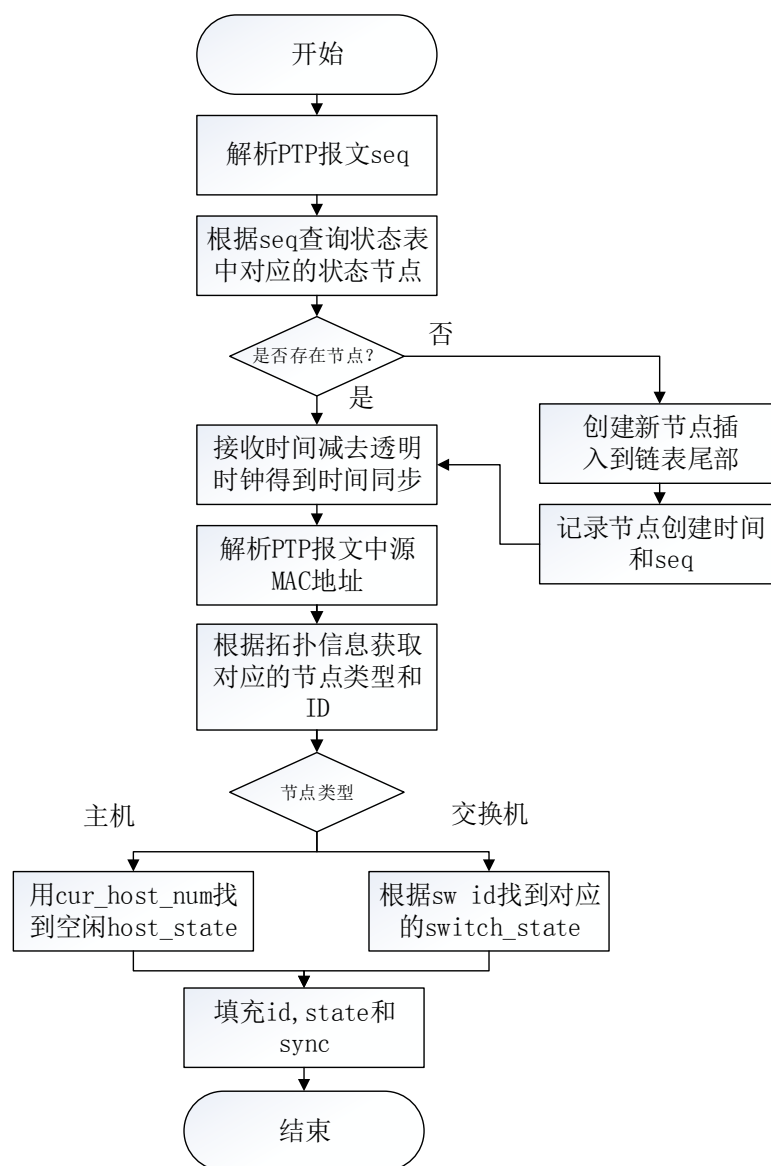


图 35 PTP 报文处理流程



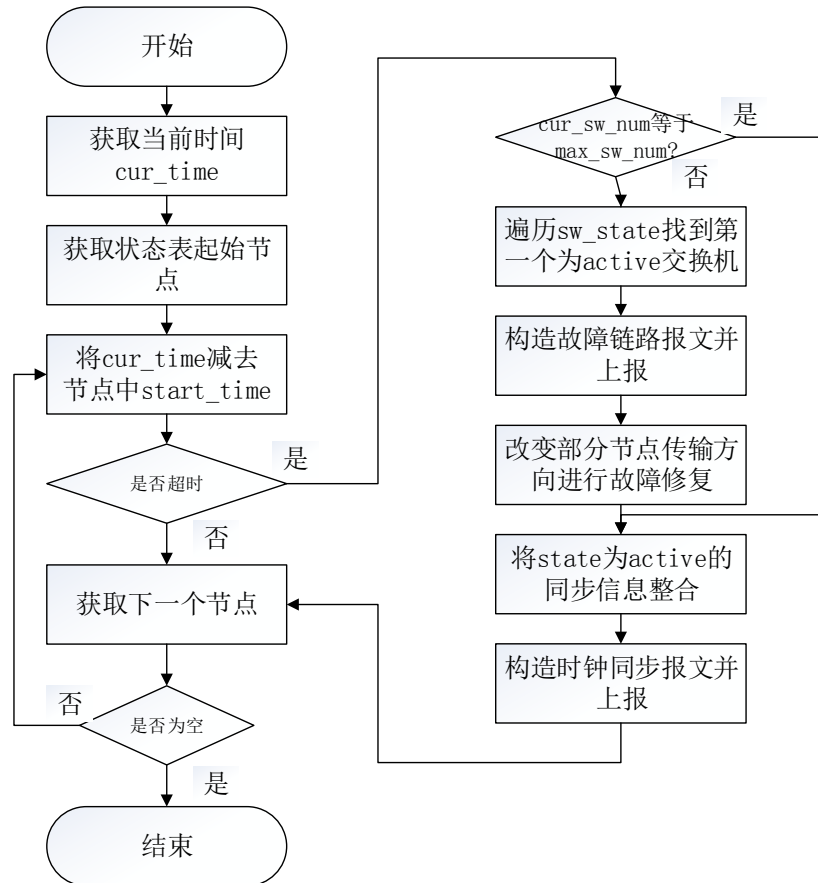


图 36 状态表处理流程

## 附录一：版本管理

文档版本号	修改人	修改时间	修改记录
V1	严锦立, 李军帅	2019.4.28	编写整个文档
V2	严锦立	2019.5.4	对总体设计、南向北向 IO 库, 拓扑管理, 资源管理和资源预约模块进行重构
V3	严锦立, 李军帅	2019.5.6	整个文档重构
V4	严锦立	2019.5.7	对南向通信子线程流程进行重新设计, 对资源预约表接口进行重新定义
V4	李军帅	2019.5.7	修改南向通信协议中 PTP 报文的部分, 没有读操作, 修改 PTP 报文格式, 添加配置信息

			到全局资源
V5	严锦立	2019.5.8	删除北向通信库中的接收处理函数，使用主动上报功能，将拓扑管理中的链路故障恢复功能放到状态监测模块中
V5	李军帅	2019.5.8	对 PTP 报文中可写可只读字段的修改，添加状态信息到全局资源管理中，维护状态信息和配置信息
V6	严锦立	2019.5.9	在资源调度模块中添加流调度信息结构体，同时更新资源调度模块的编程接口
V6	李军帅	2019.5.9	修改南向协议中的接口函数，在全局管理中删除状态信息，修改接口函数
V7	李军帅	2019.5.15	修改 PTP 的报文格式，把可读可写的部分填在 PTP 报文的末尾
V7	严锦立	2019.5.16	修改拓扑管理和资源预约文件的格式定义
V8	严锦立	2019.5.17	修改资源调度模块中 TSN 调度信息和带宽预约调度信息的数据结构定义
V8	李军帅	2019.5.17	修改 PTP 报文中配置信息的具体字段的位宽，删除映射结构体中的类型字段。
V9	李军帅	2019.5.18	修改整体处理流程的描述部分，增加了详细描述，在有写入文本和从文本读取的模块单独增

			加文本格式的章节。
V9	李军帅	2019.5.19	修改整体处理流程图变成竖行，修改南向通信、拓扑管理和全局资源管理中的流程图，都变成竖行。