

TSN insight 软件设计文档

(V1.0)



*Open***TSN**

一、设计目标

本文档的目的是通过接受到的数据，在界面上进行展示，并根据接收数据的变化，能够在界面上进行动态表现。

二、总体设计

2.1 整体架构

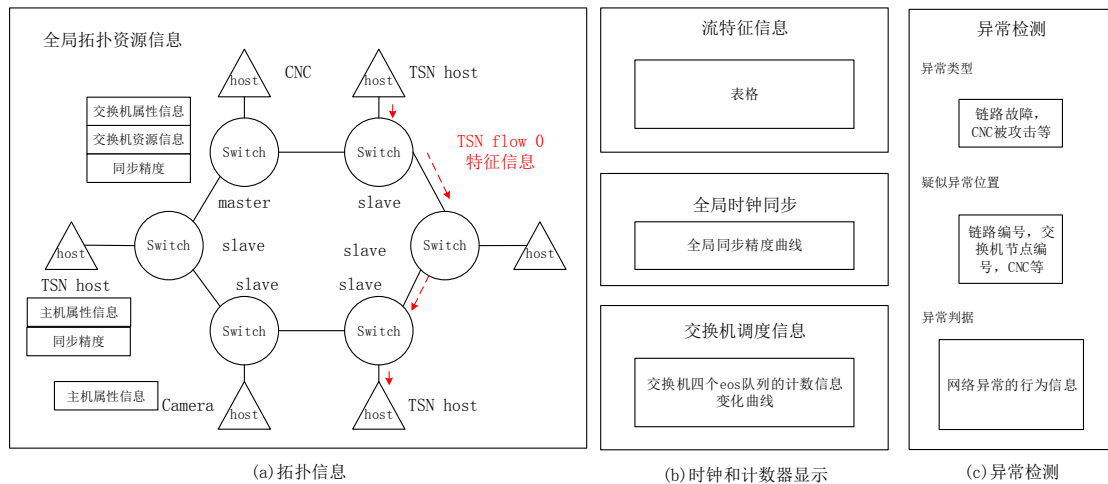


图 1 整体架构

图 a 中显示拓扑信息，展示交换机的属性和资源信息，交换机属性中包括本地 mac 地址和交换机的 sw_id，资源信息包括 CQF 队列，带宽资源等，同时展示时间同步的偏差。主机展示的时间同步的偏差。

图 b 中显示流特征信息，包括 TSN 流和带宽预约流的流特征，包括源 MAC，目的 MAC 等，其中 TSN 流还包括时间周期，最大延迟等，带宽预约流包括占用带宽的资源

图 c 为异常检测模块。

2.2 整体处理流程

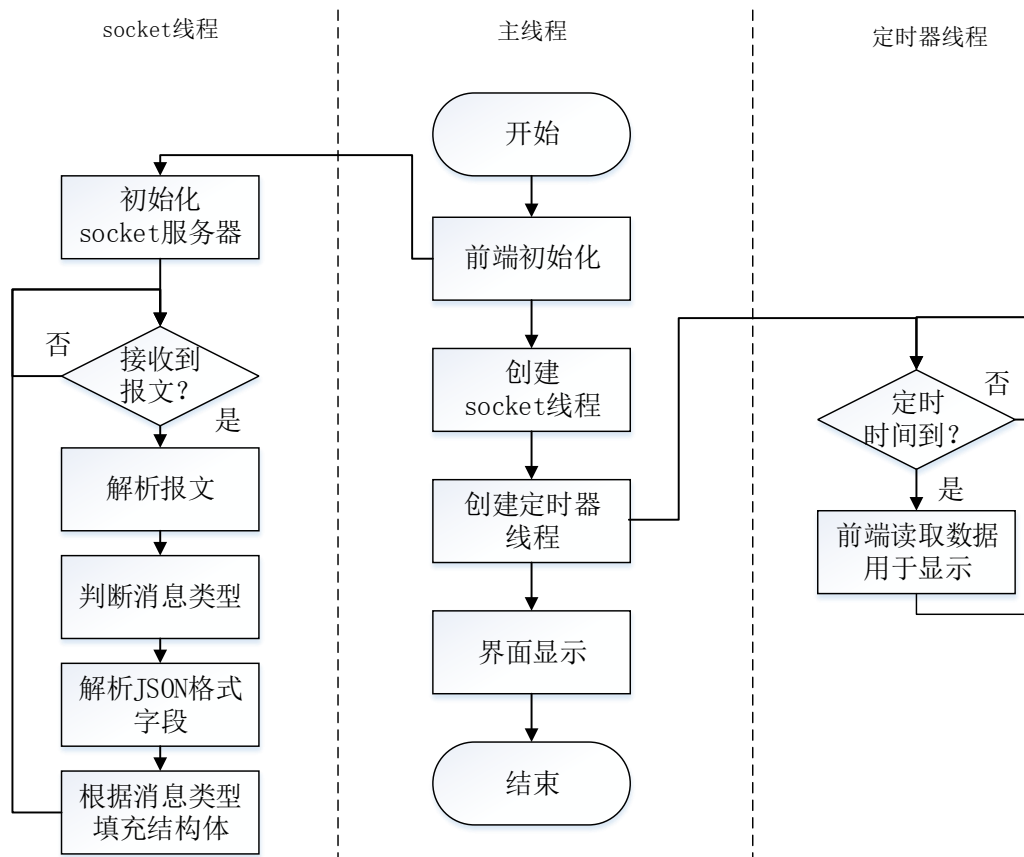


图 2 整体处理流程

采用单进程多线程架构，包括主线程，socket 服务器接收子线程和定时器子线程。

● 主线程

- (1) 前端初始化。
- (2) 初始化 socket 通信，建立 socket 服务器，接收客户端发送过来的消息，并对消息进行解析。
- (3) 初始化定时器，定时时间到后读取数据，显示在界面上。

● socket 服务器接收子线程

- (1) 该线程是一个接收线程，用于接收客户端发送的消息；
- (2) 解析消息类型，目前有拓扑信息，资源信息，流特征信息，时间同步信息，交换机计数信息。目前只支持这几种消息的解析，如果不是这几种消息，则执行 (1)。
- (3) 解析消息 JSON 格式的字段。
- (4) 把消息字段存入到结构体中，执行 (1)。

● 定时器子线程

- (1) 启动定时器，如果超时则说明定时时间到，执行 (2)；
- (2) 从内存中读取数据，执行 (1)。

三、详细设计

3.1 后端设计

3.1.1 功能描述

后端主要完成接收 CNC 发送过来的 JSON 格式的 PTP 报文，并进行解析，保存在一个结构体里面，并且提供一个接口函数供前端获取结构体里面的内容。后端的设计主要分为两部分，包括信息接收模块，信息存储查找模块。

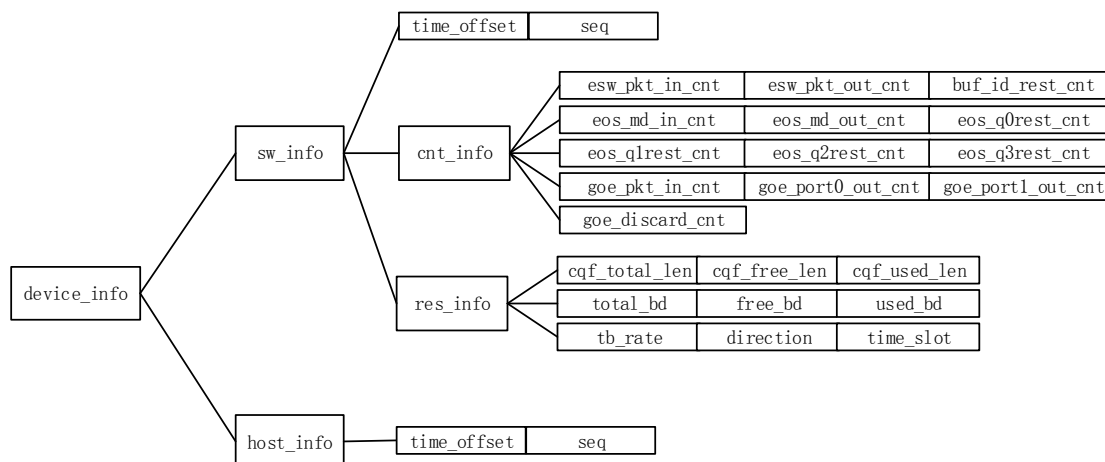
- (1) **信息接收模块：**该模块主要完成接收 CNC 发送的 JSON 格式的报文，使用 TCP socket 进行接收，该模块当做一个 socket 服务器，CNC 主动发送报文，该模块负责接收。信息接收模块在 Qt_recv.cpp 中实现，相关的数据结构在 Qt_recv.h 中定义。
- (2) **信息存储查找模块：**该模块是解析 JSON 格式的报文，通过判断信息的类型来区分是具体的消息，然后存储在对应的结构体中，并且每次接收到新的数据时都会替换相应的结构体内容。信息存储模块主要是完成需要存储的数据结构的定义，在 Qt_info_io.h 中实现，该文件中声明三个外部结构体变量，socket 接收到的数据可以存储在该结构体变量中，前端根据该结构体变量的内容显示在界面上。

3.1.2 数据结构

设备信息包含交换机信息和主机信息，

交换机信息：在交换机信息结构体数组中，数组的下标代表交换机的 sw_id，每个交换机包含时间同步偏移，资源信息和计数器信息，资源信息包含 cqf 队列，带宽资源和一些状态信息等，计数器信息表示 esw，eos，goe 等一些从硬件读上来的计数器的信息。

主机信息：主机信息结构体数组，数组的下标代表交换机的 host_id，主机信息目前只有时间同步偏移量和序列号。



拓扑信息结构体

struct topology_info		
字段	位宽	含义
sw_id	8	交换机 ID
sync_type	8	设备同步类型 master/slave
host_id	8	主机 ID
sw_mac[6]	8	交换机 MAC
host_mac[6]	8	主机 MAC
next_sw_port	8	下一个交换机的 ID
prev_sw_port	8	上一个交换机的 ID
next_poet	8	连接下一个交换机的当前交换机端口
prev_port	8	连接上一个交换机的当前交换机端口

资源信息结构体

struct resource_info		
字段	位宽	含义
cqf_total_len	32	CQF 队列的总长度
cqf_free_len	32	CQF 队列的剩余长度
cqf_used_len	32	CQF 队列的已用长度
total_bd	32	总带宽
free_bd	32	剩余带宽
used_bd	32	已用的带宽
tb_rate	32	令牌桶的速率
direction	8	传输方向
time_slot	32	时间槽的大小

TSN 流特征信息结构体

struct tsn_feature		
字段	位宽	含义
dst_sw_id	8	
flow_type	8	流类型, tsn 流
flow_id	16	流 ID
src_mac[6]	8	源端系统 MAC 地址
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
path_len	8	路径长度

direction	8	传输方向:1 为正向(1 口出, 0 口进), 0 为反向
src_sw_id	8	路径起始的交换机 ID
dst_sw_id	8	路径终点的交换机 ID

带宽预约流特征信息结构体

struct bd_feature		
字段	位宽	含义
dst_sw_id	8	
flow_type	8	流类型, tsN 流
flow_id	16	流 ID
src_mac[6]	8	源端系统 MAC 地址
dst_mac[6]	8	目的端系统 MAC 地址
priority	8	优先级 (3,4)
bandwidth	8	时间周期
path_len	8	路径长度
direction	8	传输方向:1 为正向(1 口出, 0 口进), 0 为反向
src_sw_id	8	路径起始的交换机 ID
dst_sw_id	8	路径终点的交换机 ID

流特征信息结构体

struct flow_feature		
字段	位宽	含义
tsN_flow	struct tsN_feature	tsN 流
bd_flow	struct bd_feature	带宽预约流

交换机计数信息结构体

struct counter_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
buf_id_rest_cnt	8	Bufm 中所剩余的 ID 计数器
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	8	EOS 模块 Q0 队列的剩余长度计数器
eos_q1rest_cnt	8	EOS 模块 Q1 队列的剩余长度计数器
eos_q2rest_cnt	8	EOS 模块 Q2 队列的剩余长度计数器
eos_q3rest_cnt	8	EOS 模块 Q3 队列的剩余长度计数器
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

交换机信息结构体数组

struct switch_info		
字段	位宽	含义
seq	64	序列号表示哪一轮上报
time_offset	64	时间同步偏移量
res_info	struct resource_info	资源信息
cnt_info	struct counter_info	计数器信息

主机信息结构体数组

struct host_info		
字段	位宽	含义
seq	64	序列号表示哪一轮上报
time_offset	64	时间同步偏移量

设备信息结构体数组

struct device_info		
字段	位宽	含义
sw_info[MAX_SW_NUM]	struct switch_info	交换机数组
hs_info[MAX_HOST_NUM]	struct host_info	主机数组

(1) 接收的消息格式

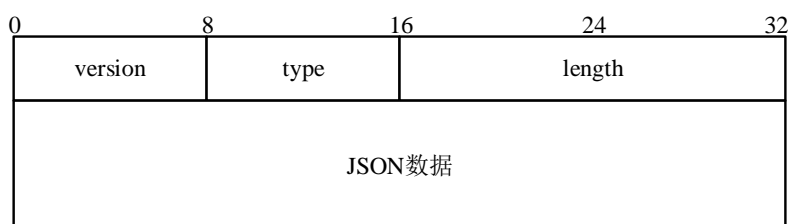


图 4 消息格式

version 版本号：当前通信协议的版本，目前为止 0；

type 消息类型

表 2 消息类型 type

类型	数值	功能
拓扑信息	0x0	上报当前网络拓扑信息，包括端系统和交换机
资源信息	0x1	上报交换机的资源信息

流特征信息	0x2	上报 TSN 流和带宽预约流的特征信息
时间同步信息	0x3	上报所有从设备相对于主设备的时间同步差值
交换机计数信息	0x4	上报每个交换机中计数器信息
链路故障信息	0x5	上报故障链路的信息

length 长度: 通用字段加上 JSON 数据的总长度

3.1.3 编程接口

(1) 初始化 socket 服务器

函数定义	int init_socket_tcp_server()
输入参数	无
返回结果	初始化成功返回 0，否则返回-1
功能描述	初始化 socket TCP 服务器，用于接收客户端发送来的报文，并把接收的内容存放在一个 buf 中

(2) 解析 JSON 格式

函数定义	int pkt_parse(u8 *data, u16 len)
输入参数	服务器接收的上报信息，长度
返回结果	解析成功返回 0，否则返回-1
功能描述	解析服务器接收到的消息内容，判断消息类型，根据消息类型调用相应数进行解析 json 格式。

(3) 解析 JSON 格式的拓扑信息

函数定义	int topology_json_parse(u8 *data)
输入参数	接收到消息的 JSON 字段
返回结果	解析成功返回 0，否则返回-1
功能描述	根据 JSON 格式的内容，解析 JSON 格式的拓扑信息，并根据解析的内容，对 struct topology_info 结构体数组进行赋值，并且 sw_id 代表数组的下标。

(4) 解析 JSON 格式的资源信息

函数定义	int resource_json_parse(u8 *data)
输入参数	接收到消息的 JSON 字段
返回结果	解析成功返回 0，否则返回-1
功能描述	解析 JSON 格式的资源信息，根据解析的 sw_id，代表 struct switch_info 结构体数组的下标，找到对应结构体，解析资源信息并对 struct resource_info 结构体进行赋值

(5) 解析 JSON 格式的流特征信息

函数定义	int flow_feature_json_parse(u8 *data)
输入参数	接收到消息的 JSON 字段
返回结果	解析成功返回 0，否则返回-1

功能描述	首先解析流特征的类型，TSN 流或者带宽预约流，如果是 TSN 流解析填充 struct tsn_feature 结构体，如果是带宽预约流，填充 struct bd_feature 结构体
------	--

(6) 解析 JSON 格式的时间同步信息

函数定义	int sync_state_json_parse(u8 *data)
输入参数	接收到消息的 JSON 字段
返回结果	解析成功返回 0，否则返回-1
功能描述	解析时间同步信息，首先解析设备类型，判断是 switch 或者 host，如果是 switch，根据 device_id 为下标，找到 struct switch_info 结构体，然后进行赋值，如果是 host，根据 device_id 为下标，找到 struct host_info 结构体，然后对 seq 和 time_offset 进行赋值

(7) 解析 JSON 格式的交换机计数信息

函数定义	int switch_counter_json_parse(u8 *data)
输入参数	接收到消息的 JSON 字段
返回结果	解析成功返回 0，否则返回-1
功能描述	解析交换机计数信息，根据 sw_id 作为下标，找到对用的 struct switch_info 结构体，然后填充 seq 和 struct counter_info 结构体。

(8) 提供交换机信息

函数定义	int get_switch_info (struct switch_info *tmp_sw, u8 sw_id)
输入参数	tmp_sw 用于保存交换机信息，sw_id 代表交换机的 id
返回结果	找到相应交换机返回 0，否则返回-1
功能描述	根据 sw_id，对应交换机结构体数组，查找到对应交换机信息，把值赋值给 tmp_sw。

3.1.4 处理流程

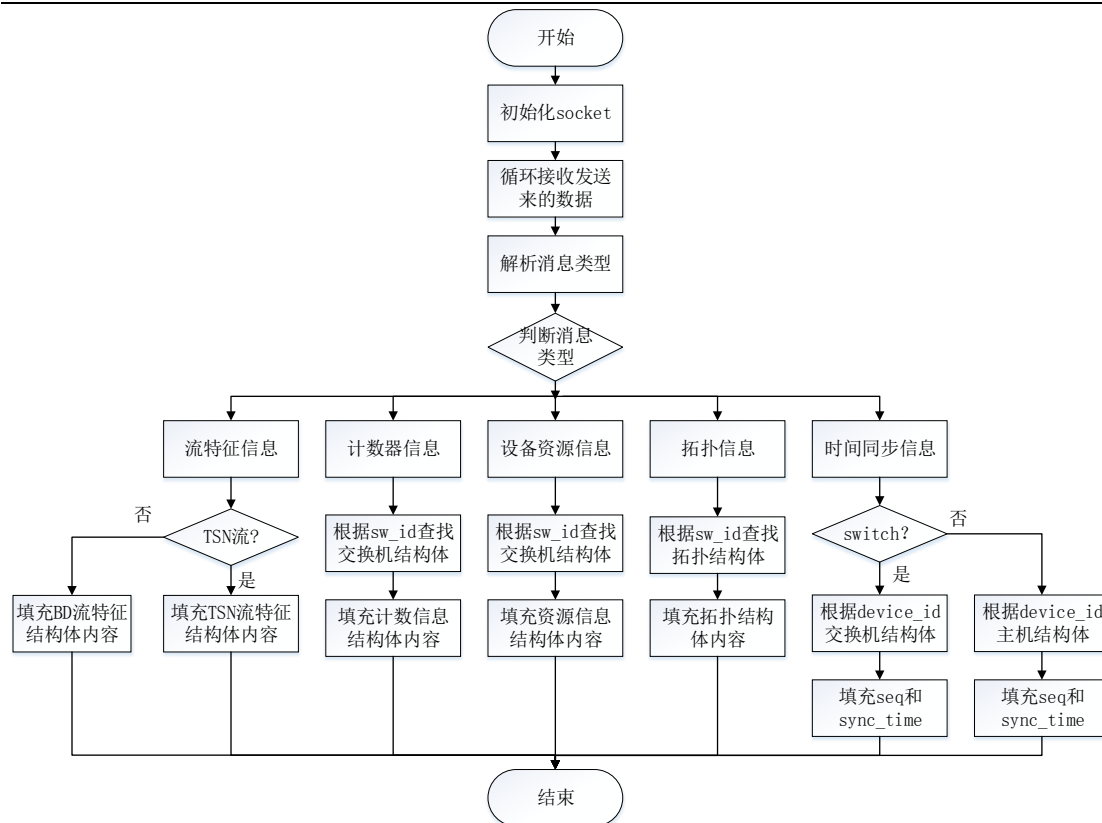


图 4 socket 接收解析流程图

附录一：版本管理

文档版本号	修改人	修改时间	修改记录
V1	李军帅	2019.6.17	编写整个文档
V2	李军帅	2019.7.15	修改整体处理流程图，删除后端设计中功能描述部分的框图，改为语言描述，修改后端设计的处理流程图，修改为按照线程的处理流程来画。