TSNSwitch 设计文档

（版本 1.6）

OpenTSN 开源项目组

2019 年 5 月

版本历史

版本	修订时间	修订内容	修订人	文件标识
1.0	**2019.05.12**	根据 **0509** 概要设计的第一版详细设计		**OPENTSN**
1.1	**2019.05.14**	根据 **0514** 讨论后文档进行修订，增加附录，对 **LCM** 进行重构，使其符合 **FAST** 流水线架构，并对应修改子模块划分和状态机设计。 将 **ESW** 模块进一步划分为三个子模块；将发往网口的分组都改为发往 **GOE** 以供 **GOE** 对分组进行计数。		
1.2	**2019.05.15**	将 **EOS** 调度解耦为 **GC** 模块和 **TS** 模块；增加 **GC** 模块详细设计； **QS** 和 **TS** 的功能不用状态机实现；画出整体框架图的所有接口信号并说明；画 **MB**、**GC**、**TS** 模块的流程图。		
1.3	**2019.06.19**	调试完成后，根据调试时作出的改动相应地调整文档。		
1.4	**2019.07.12**	修改文档格式		
1.5	**2019.11.14**	对整个文档进行重构		
1.6	**2019.12.02**	整理整篇文档		

目录

1. 项目简介

本文档是 OpenTSN 开源项目中的子项目 TSNSwitch 的设计文档，包含 TSNSwitch 设计目标、概要设计和详细设计几个主要部分。

1.1 TSNSwitch 的设计目标

TSNSwitch 是 OpenTSN 项目下的一个子项目，其设计目标主要是实现一个满足 TSN 网络传输的交换机相关功能开发。具体功能如下：

1. 使用 1588 ptp 协议进行全网的时间同步；
2. 基于 CQF 的输出调度机制；
3. 基于令牌桶的 P3-P5 优先级的流量预留调度；
4. 支持本地 TSN 节点对本地状态通过 beacon 消息进行周期性上报；
5. 提供 TAP 口进行镜像用于分析；
6. 支持环形拓扑、星形拓扑、线性拓扑等多种拓扑结构。

1.2 TSNSwitch 的设计方案

1.2.1 使用平台

TSNSwitch 基于 FAST 架构设计实现，FAST 架构如下图 1 所示：

图1 FAST 架构图

由于 TSNSwitch 对接口的处理需求与标准的 FAST 需求不一致，因此重新定制了一个 FPGA_OS，具体的接口信息可参考附录一。

TSNSwitch 整个逻辑开发是在 openbox 上进行的，该设备支持 FAST 的架构设计，openbox 的外观如下图 2 所示，openbox 有 0-3 总共 4 个 1Gbps 数据网口。

图2 Openbox 外观图

1.2.2 需求分析

TSN 网络的典型拓扑场景包括线性、环形、星形、树形等，本文档针对上述拓扑需求设计了两套架构：环形架构与非环形架构。

环形架构：0 号和 1 号口作为 TSN 网络内部连接口；2 号口作为设备口外接终端系统等等；3 号口作为 TAP 口进行镜像用于分析。

如下图 3 是环形的拓扑，各节点的 0、1 号数据网口支持确定性转发；网络中的数据由 0 号口进、1 号口出，也可以由 1 号口进、0 号口出，这需要通过 TSNSTM 进行配置。

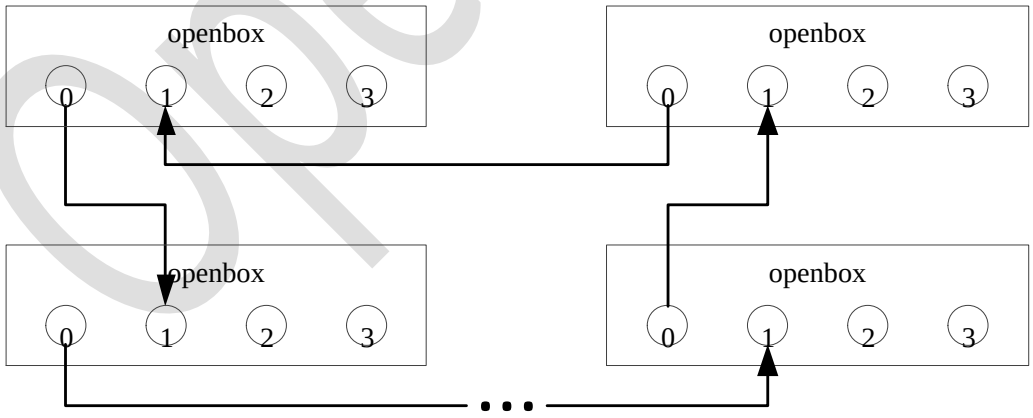


图3 环形拓扑图

非环形形架构：0 号、1 号、2 号以及 3 号口都能够作为 TSN 网络内部连接口，也能够作为设备口进行外接终端系统等等；同时提供了 TAP 功能选择，用户可通过配置 3 号口的 TAP 功能来收集交换机 0、

1、2 号接口输出流量的镜像用于分析。需要注意的是，在使用该方案搭建拓扑时网络中是不允许存在环形结构的，原因是当存在环形时，ptp 同步 sync（广播）报文就会在环内一直循环，导致在主时钟的一个同步周期内从时钟节点收到多个同步 sync 报文，因此同步逻辑就会乱掉。

如下图 4 是星形的拓扑，各节点的 4 个数据网口都需要确定性转发的功能；数据的流向由 TSNSTM 进行规划，可有多种不一样的路线进行数据流的传输。

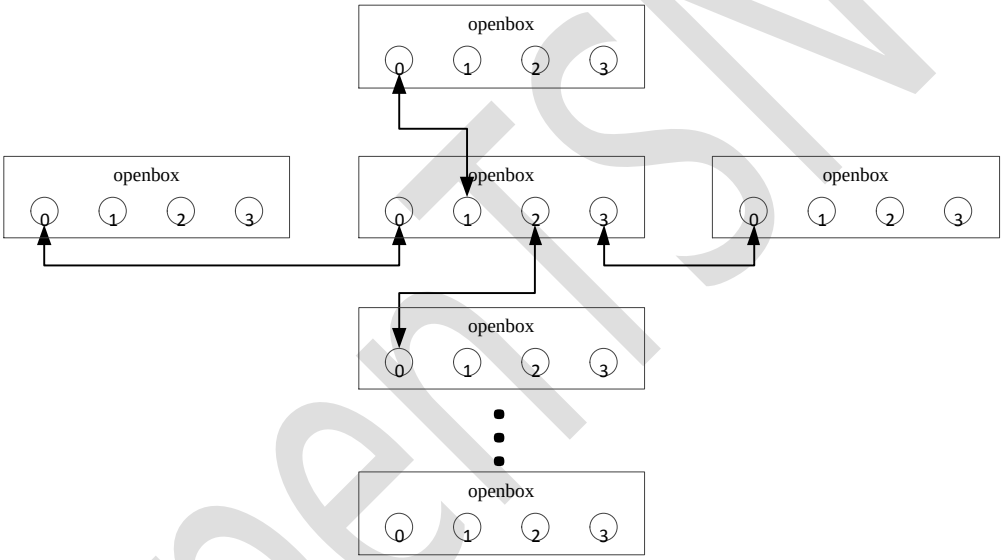


图4 非环形拓扑图

2. 概要设计

2.1 总体架构

TSNSwitch 提供了两套总体架构，分别对应环形拓扑以及非环形拓扑，下面进行逐一介绍。

2.1.1 环形拓扑架构设计

如 1.2.1 节所述，TSNSwitch 采用了 FATS 架构进行设计，但是针对 TSNSwitch 的需求，简化了控制通路的设计，TSNSwitch 架构如下图 5 所示，白色模块为 TSNSwitch 需增加的关键模块，需要重新开

发，橙色模块为对 TSNSwitch 功能进行支持需要部分修改的模块。

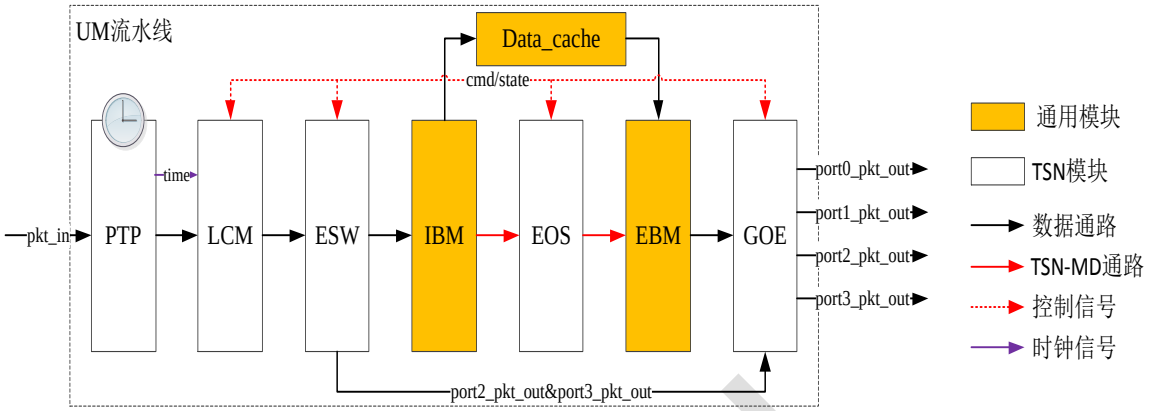


图5 环形拓扑的 TSNSwitch 总体架构设计图

其中关键模块包含以下四个部分：

PTP（Precise Time Protocol）模块：是 TSNSwitch 的时间同步模块，使用 1588ptp 协议进行全网的时间同步，具体设计参考文档“TSNSwitch 时钟同步设计文档”。

LCM（Local Control Management）模块：是 TSN 交换节点本地的管理与控制模块。主要负责维护各模块控制相关寄存器的维护与请求、周期性设备状态信息上报、处理来自 TSNSTM 节点的寄存器修改请求。

ESW 模块（Epipe SWitching）：是 TSN 交换节点的交换模块。主要负责分组类型解析、流量监管、转发动作生成以及 TS_metadata（详见附录二）生成的操作。该模块将进入流水线的分组区分为四类：控制相关（ptp）分组、TS 分组、预约带宽（RC）分组以及 best effort（BE）分组。在流水线中 BufM 资源紧张可以引起丢包时，根据上述优先级丢弃低优先级的报文保证 TS 分组的可靠传输。另外，ESW 还需要根据 TSNSTM 的配置确定所有分组的目的端口。

EOS（Epipe Output Scheduling）模块：是 TSN 节点的输出队列调度模块，用于实现简化的 cqf 转发。主要负责支持核心的

TS_metadata 的循环队列转发功能，确保 TS 分组调度输出的确定性延时，以及预约带宽分组的基于令牌桶的流量整形功能的实现。

GOE（General Output Engine）模块：主要负责对接收到的来自 EBM 的分组按照 FAST_metadata 中 “output”字段（具体参考附录一）与来自 ESW 的分组进行发送，并对发往所有发送报文进行统计计数。

处理流程：第一级模块为 PTP 模块，PTP 模块根据分组的协议字段以及分组目的 MAC 来区分分组，若需要处理的分组则在 PTP 模块内部逻辑进行处理，其他分组则旁路掉。第二级模块为 LCM 模块，LCM 根据分组的协议字段以及分组目的 MAC 来区分分组，将 beacon 配置报文里的信息提取出来并填写到对应的寄存器，其他报文则旁路掉。第三级模块为 ESW 模块，ESW 模块根据分组的分组源 MAC、输入端口号、目的 MAC 来区分分组，将分组进行往下级模块转发、往 GOE 模块转发、丢弃三种处理。第四级模块为 IBM 模块，IBM 模块负责将分组数据写入到 data_cache 模块的 RAM 中去，并从 data_cache 得到存放的 ID 号，将 ID 存放到 TS_metadata 中并传输给下级模块。第五级模块为 EOS 模块，EOS 根据 TS_metadata 中的字段信息将 TS_metadata 分为 3 种类型并根据当前时间片信息写入到 4 种队列中去，再采用 CQF 队列转发机制对 TS_metadata 进行调度。第六级模块为 EBM 模块，EBM 模块根据 EOS 模块给出 TS_metadata 中的 ID 号往 RAM 进行分组数据的读取。第七级模块为 GOE 模块，将分组根据输出端口进行区分并传输。Data_cache 模块，使用 RAM 对分组进行缓存管理，并使用 fifo 对 RAM 块的地址进行管理，分组在 RAM 进行缓存之后拿到存放的 ID 号，再根据 ID 号去对应的 RAM 地址进行分组的读取。

2.1.2 非环形拓扑架构设计

为了支持多样化的 TSN 网络拓扑，需要设计区别于环形结构的支持数据流双向确定传输的 TSNSwitch。

对于 TSN 交换节点，为了支持多样化拓扑结构，需要将 4 个网口接口都用于 TSN 网络中，同时需要保留原来 TAP 口功能。4 个网络接口都具有确定性转发功能，可用于 TSN 网络内部的连接也可外接设备。考虑到实际场景中会有对组播报文的需求，因此添加对组播报文的转发处理逻辑。

TSNSwitch 在非环形拓扑的总体架构设计如下图 6 所示，蓝色模块为对 TSNSwitch 功能进行支持需要部分修改的模块。

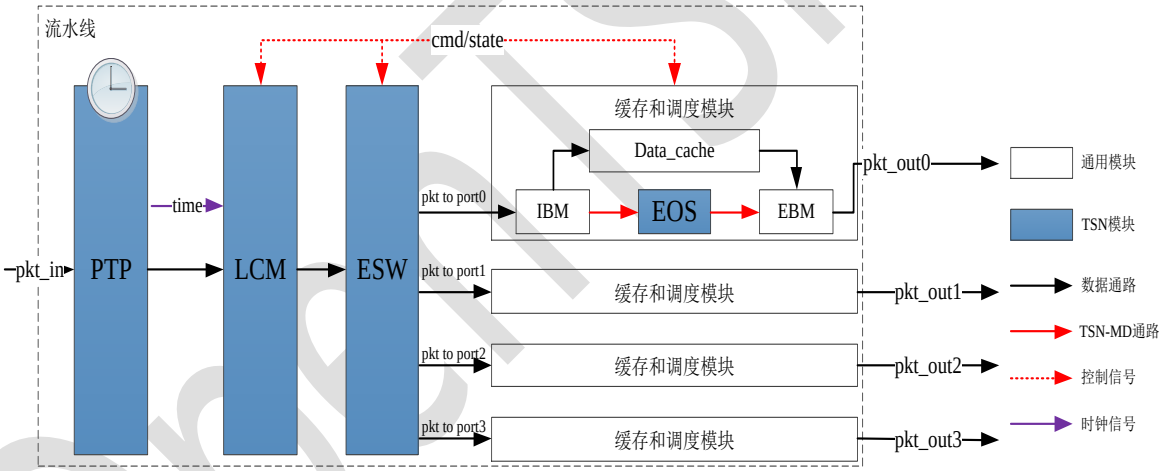


图6 非环形拓扑的 TSNSwitch 总体架构设计图

其中关键模块包含以下五个部分：

PTP 模块：同环形拓扑的架构设计方案一致。

LCM 模块：同环形拓扑的架构设计方案一致。

ESW 模块：是 TSN 交换节点的交换模块。主要负责分组类型解析、流量监管、TS_metadata（详见附录二）生成、查找转发表并将分组传入端口对应的缓存和调度模块的操作。该模块将进入流水线的分组区分为四类：控制相关分组、TS 分组、预约带宽分组以及 best

effort 分组。在流水线中 BufM 资源紧张可以引起丢包时，根据上述优先级丢弃低优先级的报文保证 TS 分组的可靠传输。

缓存和管理模块：此模块包括 IBM（Input Buffer Management）、data_cache、EOS、EBM（Extract Buff Management）模块。总共 4 个缓存和管理模块，每一个模块对应一个端口，分组数据在 data_cache 中缓存，EOS 模块根据分组的 TSN_metadat 进行调度和管理。

EOS 模块：同环形拓扑的架构设计方案一致。

处理流程：第一级模块为 PTP 模块，PTP 模块根据分组的协议字段以及分组目的 MAC 来区分分组，若需要处理的分组则在 PTP 模块内部逻辑进行处理，其他分组则旁路掉。第二级模块为 LCM 模块，LCM 根据分组的协议字段以及分组目的 MAC 来区分分组，将 beacon 配置报文里的信息提取出来并填写到对应的寄存器，其他报文则旁路掉。第三级模块为 ESW 模块，ESW 模块根据分组的分组目的 MAC、输入端口号进行查找转发表，根据查找之后得到输出端口号传输到对应的下级模块。第四级模块为 IBM 模块，IBM 模块负责将分组数据写入到 data_cache 模块的 RAM 中去，并从 data_cache 得到存放的 ID 号，将 ID 存放到 TS_metadata 中并传输给下级模块。第五级模块为 EOS 模块，EOS 根据 TS_metadata 中的字段信息将 TS_metadata 分为 3 种类型并根据当前时间片信息写入到 4 种队列中去，再采用 CQF 队列转发机制对 TS_metadata 进行调度。第六级模块为 EBM 模块，EBM 模块根据 EOS 模块给出 TS_metadata 中的 ID 号往 RAM 进行分组数据的读取。第七级模块为 GOE 模块，将分组根据输出端口进行区分并传输。Data_cache 模块，使用 RAM 对分组进行缓存管理，并使用 fifo 对 RAM 块的地址进行管理，分组在 RAM 进行缓存之后拿到存放的 ID 号，再根据 ID 号去对应的 RAM 地址进行分组的读取。

2.1.3 两种拓扑架构设计区别

2.1.3.1 整体拓扑的区别

环形拓扑的架构设计只支持所有节点组成一个环形的拓扑；非环形拓扑的架构设计支持除了包含环形的拓扑之外的所有拓扑结构，例如星形拓扑、线性拓扑结构等等。

2.1.2.2 端口的区别

环形拓扑的架构设计只有 0 号和 1 号两个端口具有确定性转发功能，而 2 号口固定作为外接设备的端口，3 号口固定作为 TAP 口。

非环形拓扑的架构设计的 4 个端口都具有确定性转发功能，4 个端口都可以进行作为外接设备的端口，没有具体约束，同时 3 号口提供选择做为 TAP 口或不做为 TAP 口两种功能。

2.1.2.3 UM 内部处理的区别

两种拓扑架构在 UM 内部的处理上主要是在于 ESW 模块上，在环形拓扑架构中 ESW 模块只将分组分为发往本节点外接设备的分组、发往 TSN 环内的分组这两种；而在非环形拓扑架构中 ESW 模块需要对分组进行查表转发表，并根据转发表的输出端口号传输到对应的下级模块中。

2.2 LCM 模块

2.2.1 LCM 模块功能

根据上节叙述，LCM 模块负责 1) 周期性读取本地所有控制相关寄存器的值，并构造包含 beacon report 消息的以太网帧进行发送；2) 解析收到的来自 TSNSTM 节点的 beacon update 消息的以太网帧，并根据对应字段值修改对应的寄存器。3）丢弃本地 LCM 发送的且未找到目的地址时返回到本地 LCM 的 beacon 报文。

2.2.2 LCM 模块概要设计

为了简化设计，在 TSNSwitch 实现中删除了原 FAST 架构的控制通路，使用 LCM 模块（本地管理模块）进行 TSN 节点全局的状态信息获取和更新。LCM 通过 beacon 报文接受远程控制器（TSNSTM）的管理控制。LCM 负责以下两个功能的实现：

1. LCM 模块周期性构造包含 beacon 消息的帧 (report 消息)，并根据附录三中定义将状态字段填写到beacon 上报报文的对应字段。
2. 而当收到来自 TSNSTM 节点的寄存器配置消息(update 消息)后，将可读写字段的值读出，并替换对应寄存器中的原始值。
3. 对进入 LCM 模块的 beacon 报文进行检测，若该 beacon 报文是本地 LCM 发出的（可根据 SMAC 地址进行判断），则将该报文进行丢弃。

LCM 模块的整体架构如图 7 所示：

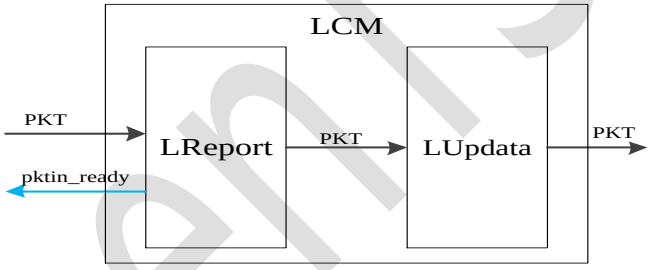


图7　LCM 模块整体架构图

LReport（LCM Report）模块：负责接收并将来自 PTP 模块的帧发送到 LUpdate 模块（由 LUpdate 模块发送至 ESW 模块）；根据全局时钟进行周期性地上报一个 beacon 报文（周期可自定义，目前方案中暂定的周期为 32ms），当时钟到达预订周期时，LReport 对 PTP 模块进行反压，并构造一个包含当前全部状态（详见附录三）信息的 Beacon Report 消息，将其发送到 LUpdate 模块中。

LUpdate（LCM Updata）模块：负责在收到来自 LReport 子模块的 Beacon Update 消息帧后，判断该消息帧是否为本地 LCM 发送的，若是则需要丢弃，若不是则利用帧数据字段包含的需更新寄存器的值更新对应的寄存器。若收到的帧非 Beacon Update 消息帧，则直接向

ESW 进行转发。

2.3 ESW 模块

2.3.1 ESW 模块功能

ESW 模块的主要功能有如下：

1） 分组类型解析；
2） MAC 地址比较；
3） 分组转发；
4） FAST_metadata 数据的修改以及 TSN_MD（TSN_metadat）数据的构造；
5） 流量监管。

2.3.2 ESW 模块概要设计

由于在环形拓扑架构与非环形拓扑架构之间 ESW 模块会有很大的不同，因此下面分两部分进行介绍。

2.3.2.1 环形拓扑架构

ESW 模块主要的功能需求为：分组类型的解析、比较 MAC 地址、FAST_metadata 的修改以及 TSN_MD 数据的构造、分组的转发、流量监管。

分组在进入 ESW 模块之后需要对分组进行解析，并将分组类型传输给 EOS 模块以供 EOS 模块根据分组的不同类型进入不同的队列；若分组的目的是本地直连设备，则直接由 ESW 模块发往 GOE 模块做计数；其他的分组则都往下级模块进行传输。往下级模块传输的分组还需根据流水线中的 bufm 空闲 ID 的多少进行分组的丢弃（仅丢弃非 TS 分组）。

针对 TAP 的需求，将所有需要进行转发的分组都复制一份往 TAP 口进行转发。

设计中将 ESW 模块设计为 3 个模块：ESW 顶层模块、PKE 模块、

PAC 模块。具体框架图以及子模块实现如下：

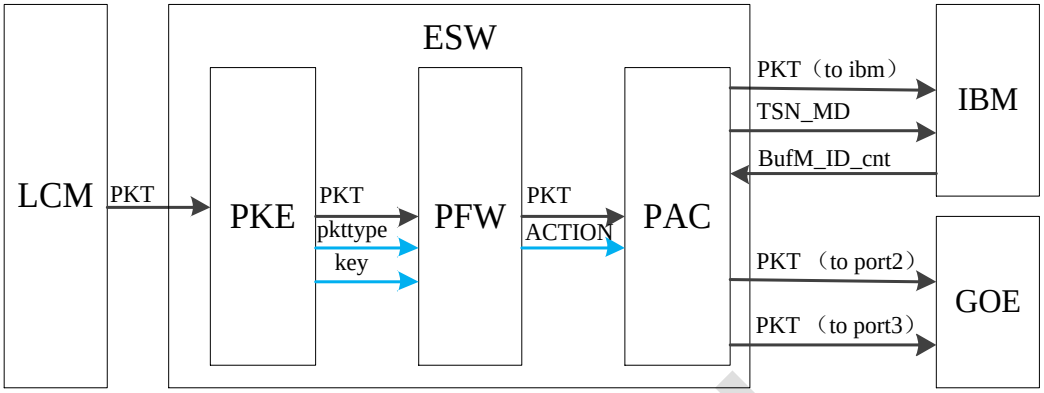


图8 环形拓扑架构的 ESW 模块划分图

图中一些关键数据的格式定义：

表1 key 格式定义

Key		
DMAC[101:54]	SMAC[53:6]	inport[5:0]

表2 ACTION 格式定义

名称	分组输出类型（pkt_outtype）	分组协议类型（pkt_type）	分组输出端口/输出 ID（outport）
位宽	2	3	6
位置	[10:9]	[8:6]	[5:0]

表3 TSN_MD 格式定义

分组协议类型[23:21]（pkt_type）	分组长度[20:9]（pkt_len）	输出端口[8]（outport）	分组存储 ID[7:0]（bufm_ID）

PKE（Pkt Key Extract）模块：判断是带 VLAN 头的分组还是标准以太网分组。若是带 VLAN 头的分组则根据 PCP 值进行分组类型的区分：TS 分组、预约带宽分组、best effort 分组；若是标准以太网分组则根据协议字段进行分组类型区分：ptp 分组、best effort 分组。并将解析的结果写到 pkttype 中；同时提取出分组的三元组{目的 MAC、源 MAC、输入端口}与分组同时传输给 PFW 模块。

PFW（Pkt Find ForWarding）模块：根据三元组的信息与本地直连设备 MAC 进行比较。源 MAC 比较：根据 3.2.3.1 节中的表项进行不同的处理。目的 MAC 比较：目的 MAC 为全 F、目的 MAC 等于本地直连设备 MAC、目的 MAC 不等于本地直连设备 MAC 三种结果。根据 3.2.3.1 节中的表项进行 ACTION 字段的修改。

PAC（Pkt ACtion）模块： 根据 ACTION 中的内容进行分组的转发。单播时：若 ACTION 的输出端口为 2 则将分组 FAST_metadata 数据根据 ACTION 进行修改之后将分组往本地直连设备传输；若 ACTION 的分组输出端口不为 2 则根据 bufm 的空闲 ID 进行分组的丢弃或转发。当 bufm 大于 3（表示 bufm 至少有 4 个空闲块可进行缓存报文），分组都可直接进行 FAST_metadata 修改之后传输分组；当 bufm 等于 3 时，分组类型为 best effort 的需要进行丢弃，其他分组进行正常传输；当 bufm 大于 0，小于 3 时，分组类型为预约带宽或 best effort 的都需要进行丢弃，其他分组进行正常传输；当 bufm 为 0 时，所有报文都进行丢弃。不丢弃时，在分组传输时需要构造 TSN_MD。广播时：处理与单播一样的处理方式之外还需要将分组复制一份往 2 号口进行转发。在所有需要进行转发的分组都复制一份转发给 TAP 口。

2.3.2.2 非环形拓扑架构

非环形拓扑架构与环形拓扑架构在整体模块设计上一样，但是各模块功能略有不同，同时在报文输出方向上也不同，下面进行详细介绍。

与环形拓扑架构设计一样将 ESW 模块设计为 3 个模块： PKE 模块、PFW 模块、PAC 模块。具体框架图以及子模块实现如下：

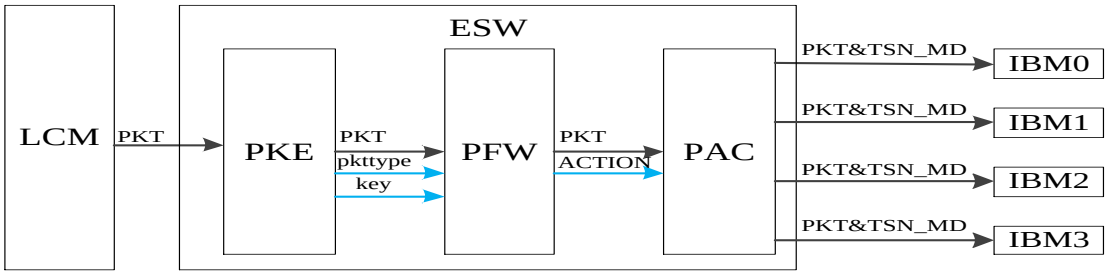


图9 非环形拓扑架构的 ESW 模块划分图

在该架构中与环形拓扑架构的关键数据基本一致，只有 key 字段略有不同，下面介绍介绍一下 key 字段的定义：

表4 非环形拓扑架构中的 key 格式定义

Key	
inport [53:48]	DMAC [47:0]

PKE 模块：判断是带 VLAN 头的分组还是标准以太网分组。若是带 VLAN 头的分组则根据 PCP 值进行分组类型的区分：TS 分组、预约带宽分组、best effort 分组；若是标准以太网分组则根据协议字段进行分组类型区分：ptp 分组、best effort 分组。并将解析的结果写到 pkttype 中；同时提取出分组的二元组{目的 MAC、输入端口}与分组同时传输给 PFW 模块。

PFW 模块：根据报文目的 MAC 地址信息区分报文是广播、组播还是单播报文；如果是单播、组播报文则查找转发表，根据查表结果得到输出端口并进行对报文的处理，转发或丢弃；如果是广播报文，则将报文的输出端口为往除了输入端口之外的三个端口。根据分组输出类型、PKE 模块解析报文时得到的报文类型信息以及输出端口，总共三个信息构造 ACTION。另外当用户选了 TAP 功能时，需要丢弃 3 口进的所有报文，同时将交换节点 0、1、2 号口传输的分组往 3 口复制一份输出。

PAC 模块：该模块负责流量监管、构造 TSN_MD，以及 TAP 模式的选择，同时将分组根据输出端口进行区分并发送给对应的 IBM 模块。当 bufm 大于 3，分组都可直接进行 FAST_metadata 修改之后传输分组；当 bufm 等于 3 时，分组类型为 best effort 的需要进行丢弃，其他分组进行正常传输；当 bufm 大于 0，小于 3 时，分组类型为预约带宽或 best effort 的都需要进行丢弃，其他分组进行正常传输；当 bufm 为 0 时，所有报文都进行丢弃。不丢弃时，在分组传输时需要构造 TSN_MD。

2.4 EOS 模块

2.4.1 EOS 模块功能

为了支持基于 FAST 的 TSN 交换，需要在 EOS 模块实现:

1. 分队列缓存不同类型的报文的 TS_metadata 数据，即使用 Q0、Q1 缓存 TSN 报文的 TS_metadata 数据，使用 Q2 缓存 ptp 和预约带宽报文的 TS_metadata 数据，以及使用 Q3 缓存 best effort 的以太网报文的 TS_metadata 数据；
2. 根据报文优先级调度队列，Q0 = Q1 > Q2 > Q3；
3. 实现 CQF 调度功能，即使用 Q0、Q1 实现乒乓队列；
4. 基于令牌桶机制对预约带宽流量（即队列 Q2 的 TS_metadata 数据）进行整形；
5. 配置令牌桶参数，目前包含两个，即令牌桶的桶深和往令牌桶中添加令牌的速率；
6. 供其他模块（LCM）读取 EOS 模块相关寄存器信息。

2.4.2 EOS 模块概要设计

EOS 模块主要的功能是实现 CQF 调度机制，同时基于令牌桶机制对 RC 分组进行整形，若 RC 分组带宽超过预约带宽，则将 RC 分组丢弃信号传给下级模块 EBM（在 EBM 模块将该 RC 分组丢弃）。

在模块中有时间片的奇偶、令牌桶的桶深和往令牌桶中注入令牌的速率三个重要参数需要 LCM 进行配置。

本节主要介绍 EOS 模块的整体架构设计，EOS 模块的整体架构如图 10 所示：

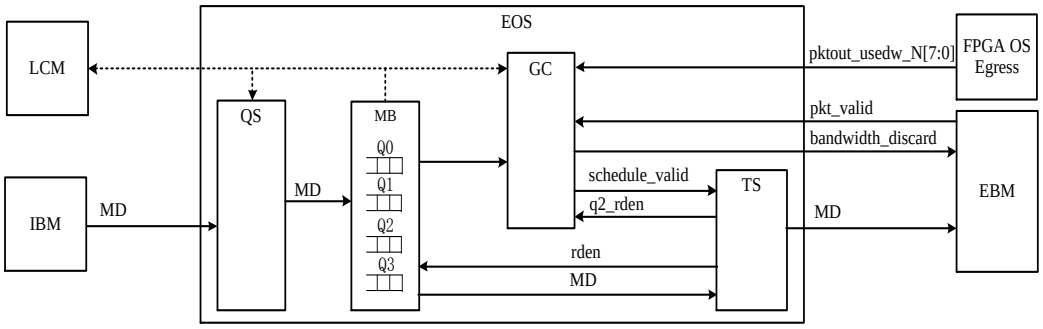


图10　EOS 模块的整体架构

QS（Queue Selecting）模块：队列选择模块，将 TS_metadata 数据写入相应的队列中。根据当前时间槽的奇偶和 TS_metadata 数据[23:21]（表示该 TS_metadata 对应的分组类型字段）决定将 TS_metadata 数据传输到哪个队列：若 TS_metadata 数据[23:21]为 3（表示该 TS_metadata 对应的分组类型为 TS 分组），且当前为奇数时间槽，则将 TS_metadata 数据[8:0]（该分组输出端口号和在 BufM 中缓存的位置 ID）传输到 Q1；若 TS_metadata 数据[23:21]为 3，且当前为偶数时间槽，则将 TS_metadata 数据[8:0]传输到 Q0；若 TS_metadata 数据[23:21]为 2（表示该 TS_metadata 对应的分组类型为 ptp 分组），则将 TS_metadata 数据[19:9]（分组长度字段）置 0（ptp 分组不需要进行流量整形，不需要消耗令牌），若 TS_metadata 数据[23:21]为 1（表示该 TS_metadata 对应的分组类型为 RC 分组），则将 TS_metadata 数据[19:9]减去两拍 FAST_metadata 长度（32 字节），并连同 TS_metadata 数据[8:0]传输给 Q2；若 TS_metadata 数据 [23:21]为 0（表示该 TS_metadata 对应的分组类型为 BE 分组），则将 TS_metadata 数据[8:0]传输给 Q3。

MB（Metadata Buffer）模块：TS_metadata 数据缓存模块，Q0 为缓存 TS 分组的 TS_metadata 数据的偶数时钟队列，Q1 为缓存 TS 分组的 TS_metadata 数据的奇数时钟队列，Q2 为缓存 RC 分组和 PTP 分组的 TS_metadata 数据队列，Q3 为缓存 BE 分组的 TS_metadata 数据队列。等待 TS 模块的 TS_metadata 数据读信号，将相应队列的 TS_metadata 数据读出。

GC（Gate Control）模块：门控模块。根据当前时间片的奇偶，Q0、Q1 队列是否为空，端口 fifo 是否有足够的空闲空间来判断 Q0、Q1 队列是否可被调度；根据 Q2 队列是否为空，端口 fifo 是否有足够

的空闲空间来判断 Q2 队列是否可被调度；根据 Q3 队列是否为空，端口 fifo 是否有足够的空闲空间来判断 Q3 队列是否可被调度；并将 Q0、Q1、Q2、Q3 队列是否可被调度的判断结果传给 TS 模块。同时基于令牌桶机制对 RC 流进行整形，决定 RC 分组是否应在下级模块 EBM 中进行丢弃。

TS（Transmiting and Scheduling）模块：调度发送模块，根据 GC 模块传来的 Q0、Q1、Q2、Q3 队列是否可被调度的判断结果，对可被调度的队列采用绝对优先级调度策略来调度 MB 模块中队列的 TS_metadata。

2.5 GOE 模块

根据 GOE 模块分析，对 GOE 模块的设计如下图：

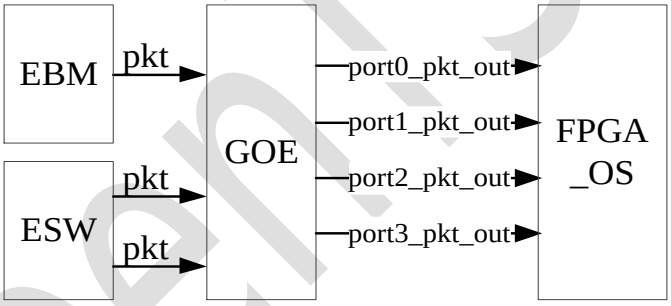


图11 GOE 模块结构图

该模块的功能比较简单，根据进入的分组判断往哪个接口进行传输，之后便将分组传输给对应的接口。同时对所有传输的分组进行计数。

2.6 交换机管理控制

TSN 网络由 TSNSTM 进行统一的配置管理，TSNSTM 对各节点通过 ptp 报文实现管理，ptp 报文通过目的 MAC 来指定需要配置的目的节点。因此各节点都需要拥有自己的一个 MAC 地址，本设计中规定了 TSN 网络节点 MAC 的前 40 位为 00：06：06：00：00，后 8 位可通过软件进行配置来区分不同的节点。

TSNSTM 知道每个节点的 MAC 地址之后，可发送 ptp 报文进行配置，节点在收到目的 MAC 为本地 MAC 的 ptp 报文之后会提取出报文中的一些特定字段信息对本地寄存器的值进行更新。

每个节点也会按周期向 TSNSTM 上报一个 ptp 报文，该报文携带着节点的一些信息，包括 BufM 使用情况、各模块的计数器等。用户可在 TSNSTM 通过对收到的 ptp 报文信息对各节点的运行状态进行了解，若一个节点或多个节点没有上报的报文可直接在 TSNSTM 便可定位问题是出现在哪个交换机上。

4. 可读写寄存器设计

TSNSTM 可配置/读写寄存器均迁移到 LCM 中实现。LCM 向其它模块提供对应的寄存器输出信号。如果维持所有寄存器作为周期性上报信息，则需要扩展 ptp 报文的数据字段，详见附录三。

在环形拓扑架构与非环形拓扑架构中，两个架构提供的可读写寄存器是不同的，下面分别进行介绍。

4.1 环形拓扑中的可读写寄存器

读写	信号名	含义	位宽
读写	Direction_mac_addr	TSN 节点直连设备 MAC 地址。	48
读写	Direction	设备口接收报文的转发方向，0 为 0 口转发，1 为 1 口转发。	1
读写	token_bucket_depth	令牌桶深度	16
读写	token_bucket_rate	令牌桶速率，控制 P3-P5 优先级流量整形	16
读写	time_slot_period	时间槽大小、时间片翻转周期。	32
只读	Esw_pktin_cnt	进入 ESW 模块的分组计数器。	64
只读	Esw_pktout_cnt	ESW 模块输出的分组计数器。	64

只读	Local_mac_addr	TSN 本地 MAC 地址低 8。	8
只读	Bufm_ID_cnt	Bufm 中所使用的 ID 计数器	8
只读	Eos_mdin_cnt	进入 EOS 模块的 TS_metadata 计数器	64
只读	Eos_mdout_cnt	EOS 模块输出的 TS_metadata 计数器	64
只读	Eos_q0_used_cnt	EOS 模块 Q0 队列已使用长度计数器	8
只读	Eos_q1_used_cnt	EOS 模块 Q1 队列已使用长度计数器	8
只读	Eos_q2_used_cnt	EOS 模块 Q2 队列已使用长度计数器	8
只读	Eos_q3_used_cnt	EOS 模块 Q3 队列已使用长度计数器	8
只读	Goe_pktin_cnt	进入 GOE 模块的分组计数器	64
只读	Goe_port0out_cnt	GOE 模块往 0 口输出的分组计数器	64
只读	Goe_port1out_cnt	GOE 模块往 1 口输出的分组计数器	64
只读	Goe_discard_cnt	GOE 模块丢弃的分组计数器	64

4.2 非环形拓扑中的可读写寄存器

读写	信号名	含义	位宽
读写	reg_tap	交换机功能选择。置为 1 代表选择 TAP 功能模式；为 0 代表非 TAP 功能模式	1
读写	Token_bucket_para	为令牌桶速率。	16
读写	Token_bucket_depth	为令牌桶桶深。	16
读写	time_slot_period	时间槽大小、时间片翻转周期。	32
读写	MAC_*N	第 N 条流表的 MAC 地址	48

		（N=0-11）	
读写	port_*N	第 N 条流表的输出端口（N=0-11）	6
只读	Esw_pktin_cnt	进入 ESW 模块的分组计数器。	64
只读	Esw_pktout_cnt*N	ESW 模块往 N 号口（N=0/1/2/3）的分组计数器。	64
只读	Time_slot_flag	奇偶时间槽，当前位于奇数时间槽为 1；偶数时间槽为 0。	1
只读	Eos_mdin_cnt*N	进入 N 号端口（N=0/1/2/3）EOS 模块的 TS_metadata 计数器	64
只读	Eos_mdout_cnt*N	N 号端口（N=0/1/2/3）EOS 模块输出的 TS_metadata 计数器	64
只读	Eos_q0_used_cnt*N	N 号端口（N=0/1/2/3）EOS 模块 Q0 队列已使用长度计数器	8
只读	Eos_q1_used_cnt*N	N 号端口（N=0/1/2/3）EOS 模块 Q1 队列已使用长度计数器	8
只读	Eos_q2_used_cnt*N	N 号端口（N=0/1/2/3）EOS 模块 Q2 队列已使用长度计数器	8
只读	Eos_q3_used_cnt*N	N 号端口（N=0/1/2/3）EOS 模块 Q3 队列已使用长度计数器	8
只读	Bufm_ID_cnt*N	N 号端口（N=0/1/2/3）Bufm 中所使用的 ID 计数器	8

附录 A　FAST 架构与 FAST_metadata 格式定义

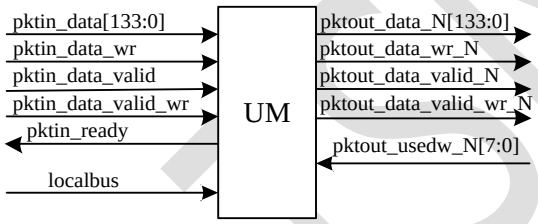
1、FAST 总体架构

FAST 总体架构可参考 FAST 官网上的“fast 原理与应用”文档。可在里面了解到 FAST 的框架以及 FPGA_OS 等。

2、TSNSwitch 与标准 FAST 之间的不同

2.1 接口的不同

TSNSwitch 中的 UM 与 FPGA_OS 的接口与标准中的不同，因此需要重新定制一套 FPGA_OS，在 TSNSwitch 中的接口定义如下：



UM 的接口定义如下：

信号名	方向	位宽	描述
FPGA OS Ingress to UM 信号定义			
pktin_data_wr	Input	1	报文数据写信号
pktin_data	Input	134	报文数据
pktin_data_valid	Input	1	报文数据标志位
pktin_data_valid_wr	Input	1	报文数据标志位写信号
pktin_ready	output	1	数据 ready 信号
Localbus	In/output	X	内部 CPU 与 UM 逻辑进行通信的 localbus 通道相关接口信号
UM to FPGA OS Egress 模块			
pktout_data_wr_N	output	1	输出报文写信号，N 为 0-3
pktout_data_N	output	134	输出报文数据，N 为 0-3
pktout_data_valid_N	output	1	输出报文标志位，N 为 0-3
pktout_data_valid_wr_N	output	1	输出报文标志位写信号，N 为

			0-3
pktout_usedw_N[7:0]	input	8	输出报文 fifo usedw 信号，N 为 0-1

在与标准 FAST 的接口定义上，主要的区别在与：

1）去除了 cin/cout 通路，在 TSNSwitch 中不再采用 cin/cout 通道进行 cpu 与 um 流水线之间的通信，um 流水线中的相关寄存器的配置都采用 beacon 机制进行。

2）分组输出信号改为 4 组信号，根据 TSNSwitch 的设计，在 TSNSwitch 的 um 流水线中以及将分组根据输出端口进行了区分，FPGA_OS 提供了 4 组分组信号，分别对应了 4 个端口。

3）将 FPGA_OS 中的 fifo 使用情况信号“pktout_usedw”代替原来的“pktout_ready”信号，为了保证时间敏感流在 EOS 进行调度之后是无障碍地从交换机传输出去，因此 EOS 之后的所有逻辑都不能存在 ready 信号，而为了保证 EOS 调度出去报文能够不丢包地传输，需要 FPGA_OS 提供一个“pktout_usedw”信号来进行控制。

2.2 FPGA_OS 的不同

1）TSNSwitch 的 FPGA_OS 也是需要重新定制的，因为前面也提到了在 TSNSwitch 中 um 流水线的所有配置都由 beacon 机制完成，因此需要去除 cin/cout 通路，那么在 FPGA_OS 中的 DMA 通道也需要删除。

2）TSN 网络中要求时间同步，实现 TSNSTM 对整个 TSN 网络的配置以及管理。在 TSNSwitch 中增加了时间同步功能，可实现设备内的时间同步以及设备间的时间同步。因此 FPGA_OS 也相应地需要增加时间同步的相关逻辑支持。

2.3 metadat 的不同

输入及输出数据分组包括 Metadata 头部及有效数据分组两部分，格式如图 37 所示，Metadata 在 FAST 报文的前 32 字节携带，每个分组进出 UM 的第 1 拍 16 字节为 Metadata0，第二拍数据为 Metadata1。

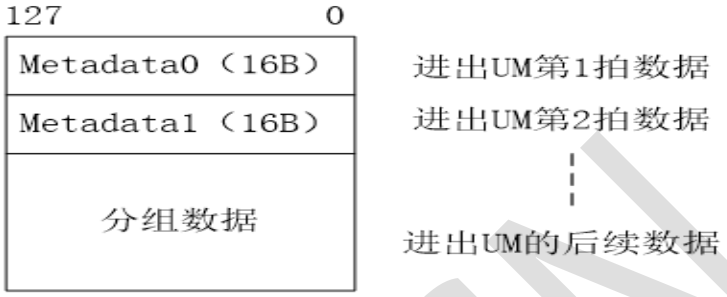


图12 分组数据传输格式

接口分组(packet)是应用在 FPGA OS 与 UM 接口上的 134bit 的数据格式，其中高 6 位为控制信息，低 128 位为报文数据。分组的前两拍为 FPGA OS 添加的 32 字节的 metadata，两拍后的数据为有效分组数据。134 位的数据由 2 位的头尾标识，4 位无效字节数，128 位的有效数据组成。

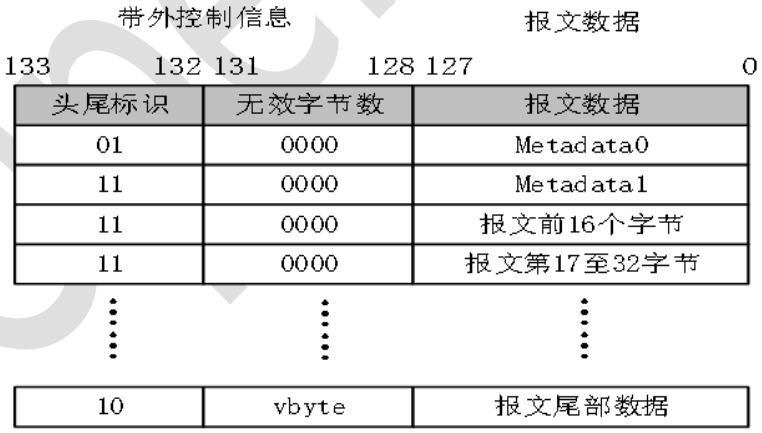


图13 报文分组传输格式

其中，[133:132]位为报文数据的头尾标识，01 代表报文头部，11 代表报文中间数据，10 代表报文尾部；[131:128]位为 4 位的无效字节数，其中 0000 表示 16 个字节全部有效，0001 表示最低一个字节无效，最高 15 个字节有效，依次类推，1111 表示最低 15 个字节无

效，最高一个字节有效。格式如图 38 所示。

下面先给出 TSNSwitch 中的 FAST_metadata 的格式定义：

表5 TSNSwitch 中的 FAST_metadata 的格式定义

字段	名称	备注
FAST_MD[127]	分组输入类型，pkttype	0: 数据报文，1: 控制报文
FAST_MD[126]	分组目的, pktdst	0: 网络接口输出，1: 送 CPU
FAST_MD[125:120]	分组输入端口号，inport	
FAST_MD[119:118]	输出类型，outtype	00: 单播 01: 组播 10: 泛洪 11:从输入接口输出
FAST_MD[117:112]	输出，output	单播：分组输出端口 ID 组播/泛洪：表地址索引
FAST_MD[111:109]	优先级，priority	分组优先级
FAST_MD[108]	丢弃位，discard	
FAST_MD[107:96]	分组长度，len	包含 metadata 字段的分组长度（按字节计算）
FAST_MD[95:88]	上次处理模块号，smid	最近一次处理分组的模块 ID
FAST_MD[87:80]	目的模块号, dmid	下一次处理分组的模块 ID
FAST_MD[79:72]	pst	标准协议类型
FAST_MD[71:64]	序列号, seq	分组接收序列号
FAST_MD[63:50]	flowid	流 ID
FAST_MD[49]	分组的来源, pktsrc	0: 网络接口输入，1: CPU 输入
FAST_MD[48]	reserve	保留
FAST_MD[47:0]	48 位时间戳, ts	时间戳

在 TSNSwitch 中的 FAST_metadat 与标准中的 FAST_metadat 的区别在于 metadat0 中的时间戳的位宽，在标准中时间戳是 32 位，而在 TSNSwitch 中时间戳是 48 位的。

附录 B　TSN_metadat 格式定义

TSN_metdata 是 TSNSwitch 特有的一个关键数据信息，主要是为了标识分组，

字段	名称	备注
TSN_MD[23:21]	分组协议类型, pkt_type	0: best effort, 1: 预约带宽，2: ptp, 3: TSN
TSN_MD[20:9]	分组长度, pkt_len	按字节数计算
TSN_MD[8]	输出端口, outport	0: 0 号端口输出，1: 1 号端口输出
TSN_MD[7:0]	分组存储 ID, bufm_ID	对应 bufM 模块地址编号

附录 C　beacon 协议与报文设计

1、Beacon 消息通信模型

在 TSNSwitch 中，我们使用 beacon 协议支持两种关键功能：

1） TSN 节点当前状态信息周期性上报；

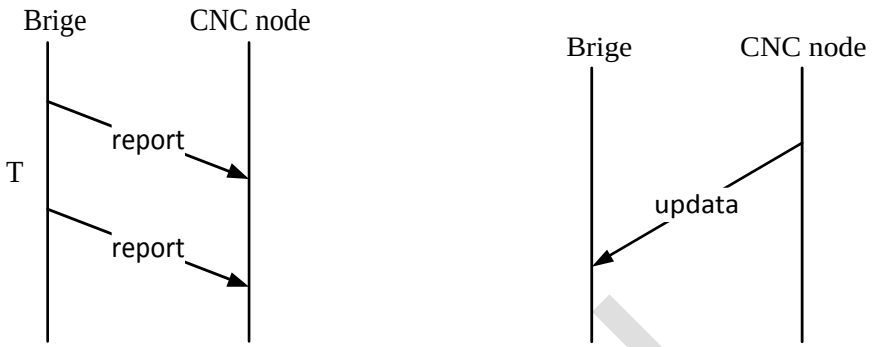
TSN 节点周期性将 TSNSTM 节点关注的本地状态信息填写到 beacon 协议报文（扩展的 ptp 报文）中，并上传 TSNSTM 节点，其中包括各队列的利用率，各端口接收与发送的分组数等等 。另外，通过周期性上报机制可对 TSN 节点是否正常运行进行检测。

2） TSNSTM 节点对 TSN 节点进行配置；

TSNSTM 节点可将 TSN 节点的需配置信息（直连设备 MAC 地址、分组转发方向、时间片大小等）写入 beacon 报文中并发送到目的端，TSN 节点接收到该报文后对本地相关寄存器进行修改。

综上，功能一可使用某个时间周期进行上报 BEACON 报文，在此设计中，上报周期为 1MS。功能二只要是对网络进行配置，在网络重启的时候或需要对网络进行重现规划时则通过 TSNSTM 进行发送配置报文。节点上报的 BEACON 报文和 TSNSTM 的配置报文可集成在同一协议中实现（借助 ptp 协议报文进行扩展）。目前的 ptp 协议

设计中保留了很多 reserved 域，可被我们用于寄存器读写的地址和数据。



（a）周期性上报场景　　（b）写寄存器场景

图14　TSNSTM-Bridge 通过 NMAC 协议的通信场景

beacon 在 TSNSwitch 中支持两种 TSNSTM-Bridge 的通信场景：周期性上报场景与写寄存器场景。周期性上报场景如图 39（a）所示，beacon 协议支持 TSN 节点对本地参数（如规则表项、计数器等）向 TSNSTM 集中控制节点以固定时间周期 T 进行上报。而写寄存器场景与图 39（b）所示，支持 TSNSTM 对 Bridge 设备的流表规则、本地 MAC 地址进行写入或配置。需要注意的是，beacon 的写报文采用“完全覆盖”的方式对寄存器进行写操作，即当 TSN 节点收到 TSNSTM 写报文时，将报文中对应的寄存器值全部替换寄存器的当前值。

2、Beacon 协议报文格式设计

2.1 ptp 协议报文格式

由于 beacon 报文是基于 ptp 协议报文进行扩展的，因此下面给出 ptp 协议报文的格式如下：

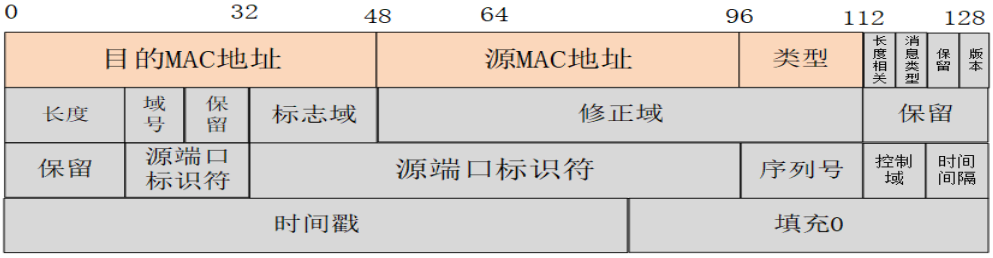


图15 ptp 同步报文格式

[注：类型：16’h88F7；

消息类型：sync 为 4’d1，delay_req 为 4’d3，delay_resq 为 4’d4，delay_test 为 4’d5；

长度为：16’d64 字节；

修正域：透明时钟，起始时，该域为 0；

时间戳：为时间戳（其他无需关系的 ptp 字段填充 0）]

2.2 环形的拓扑设计方案中的 beacon 报文格式：

本节介绍环形的拓扑设计方案中的beacon 协议报文格式的详细设计。beacon 协议基于扩展的 ptp 协议报文实现，其定义如图 41 所示。其中黄色字段为 ptp 协议使用字段，绿色和蓝色部分为数据扩展字段。数据扩展字段从第 65 字节起始（FAST 传输报文数据的第 5 拍）。根据第四章读写寄存器详细设计，需要上报的状态信号与 ESW, EOS 与 GOE 三个模块相关。

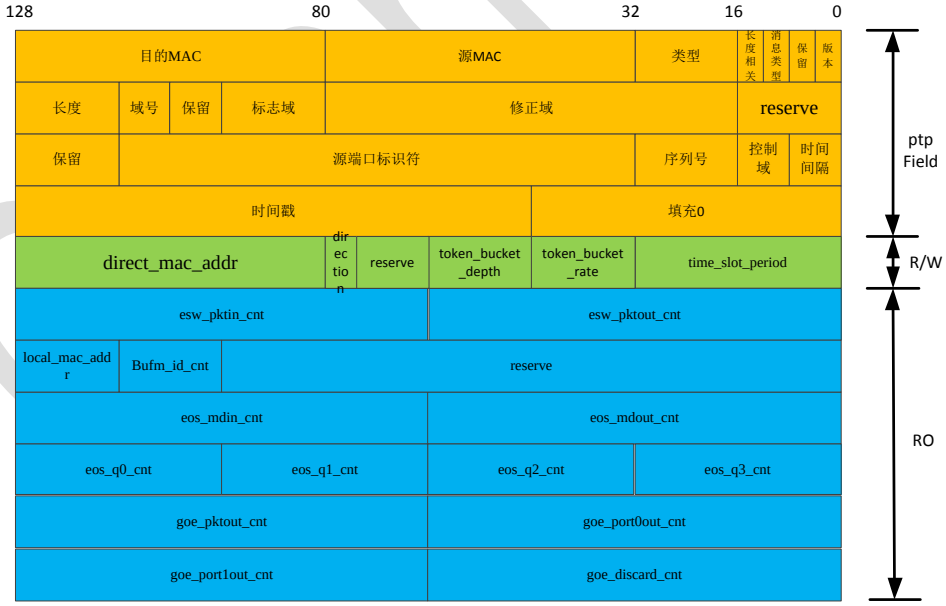


图16　beacon 报文格式定义

beacon 报文中的各字段的具体含义由下表 35 进行说明，由于前 4 拍是 ptp 报文的协议字段，并没有进行大的修改，只修改了第一拍的“消息类型”字段，当该字段为“0xE”时为 TSNSTM 配置报文，当该字

段为“0xF”时为 beacon 上报报文。而从第 5 拍开始就是 beacon 报文的扩展字段，具体拍数的字段含义如下：

表6 Beacon Report/Update 消息域划分表

拍数	位置	字段	含义
5 (读写)	[127:80]	direct_mac_addr	TSN 节点直连设备 MAC 地址。
	[79]	direction	设备口接收报文的转发方向，0 为 0 口转发，1 为 1 口转发.
	[63:48]	token_bucket_depth	令牌桶深度
	[47:32]	token_bucket_rate	令牌桶速率，控制 P3-P5 优先级流量整形
	[31:0]	time_slot_period	时间槽大小、时间片翻转周期。
6 (只读)	[127:64]	esw_pktin_cnt	进入 ESW 模块的分组计数器。
	[63:0]	esw_pktout_cnt	ESW 模块输出的分组计数器。
7 (只读)	[127:120]	local_mac_addr	TSN 本地 MAC 地址。
	[119:112]	bufm_id_cnt	Bufm 中所使用的 ID 计数器
8 (只读)	[127:64]	eos_mdin_cnt	进入 EOS 模块的 TS_metadata 计数器
	[63:0]	eos_mdout_cnt	EOS 模块输出的 TS_metadata 计数器
9 (只读)	[127:120]	eos_q0_used_cnt	EOS 模块 Q0 队列已使用长度计数器
	[119:112]	eos_q1_used_cnt	EOS 模块 Q1 队列已使用长度计数器
	[111:104]	eos_q2_used_cnt	EOS 模块 Q2 队列已使用长度

			计数器
	[103:96]	eos_q3_used_cnt	EOS 模块 Q3 队列已使用长度计数器
10 (只读)	[127:64]	goe_pktin_cnt	进入 GOE 模块的分组计数器
	[63:0]	goe_port0out_cnt	GOE 模块往 0 口输出的分组计数器
11 (只读)	[127:64]	goe_port1out_cnt	GOE 模块往 1 口输出的分组计数器
	[63:0]	goe_discard_cnt	GOE 模块丢弃的分组计数器

2.3 非环形的拓扑设计方案中的 beacon 报文格式：

在非环形的拓扑设计中的 beaocn 报文格式具体的报文格式如下：（ptp Field：ptp 帧的域；R/W：可读可写字段；RO：only read 只读字段）

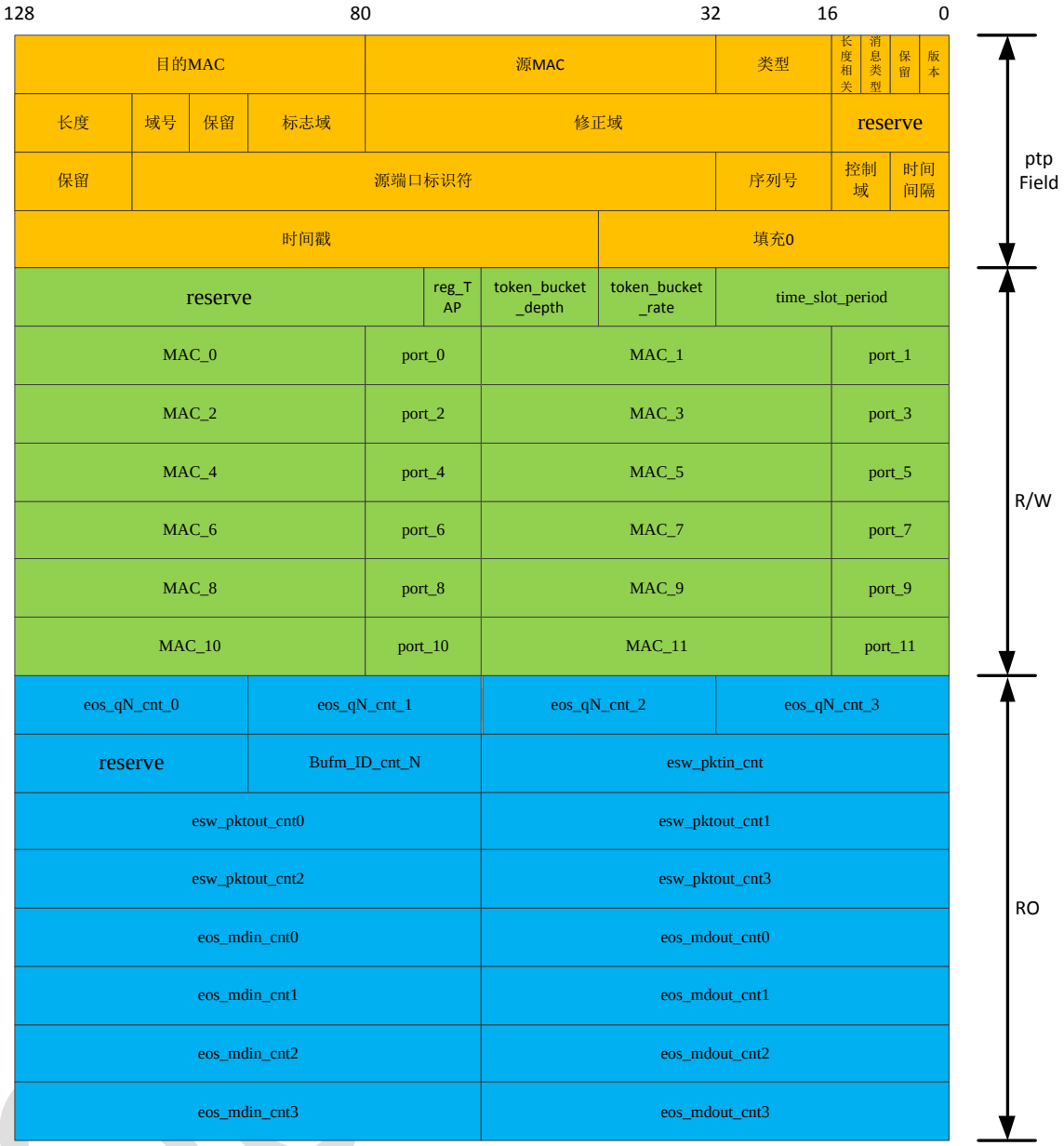


图17 Beacon 报文格式定义

beacon 报文中的各字段的具体含义由下表 35 进行说明，由于前 4 拍是 ptp 报文的协议字段，并没有进行大的修改，只修改了第一拍的“消息类型”字段，当该字段为“0xE”时为 TSNSTM 配置报文，当该字段为“0xF”时为 beacon 上报报文。而从第 5 拍开始就是 beacon 报文的扩展字段，具体拍数的字段含义如下：

表7 Beacon Report/Update 消息域划分表

模块	拍数	位置	字段	含义
LCM	5 (读写)	[127:65]	reserve	保留
		[64]	reg_tap	交换节点 TAP 功能的选择： 0：非 TAP 功能模式 1：TAP 功能模块
		[63:48]	token_bucket_depth	令牌桶深度
		[47:32]	token_bucket_rate	令牌桶速率，控制 P3-P5 优先级流量整形
		[31:0]	time_slot_period	时间槽大小、时间片翻转周期。
ESW	6-11 (读写)	[127:80]	MAC	转发表的表项（MAC 地址）
		[79:64]	Port	转发表的动作（输出端口）
		[63:16]	MAC	转发表的表项（MAC 地址）
		[15:0]	Port	转发表的动作（输出端口）
EOS	12 (只读)	[127:96]	eos_qN_cnt_0	端口 0 的 EOS 模块队列使用长度计数，（N=0/1/2/3，对应 4 个队列）
		[95:64]	eos_qN_cnt_1	端口 1 的 EOS 模块队列使用长度计数，（N=0/1/2/3，对应 4 个队列）
		[63:32]	eos_qN_cnt_2	端口 2 的 EOS 模块队列使用长度计数，（N=0/1/2/3，对应 4 个队列）
		[31:0]	eos_qN_cnt_3	端口 3 的 EOS 模块队列使用长度计数，（N=0/1/2/3，对应 4 个队列）
	13 (只读)	[127:96]	reserve	保留
		[95:64]	BufM_ID_cnt_N	缓存模块 BufM 的使用的 ID 计

ESW				数。（N=0/1/2/3，对应 4 个端口）
		[63:0]	esw_pktin_cnt	进入 ESW 模块的分组计数器
	14 (只读)	[127:64]	esw_pktin_cnt0	进入 GOE 模块的分组计数器
		[63:0]	esw_pktin_cnt1	GOE 模块往 0 口输出的分组计数器
	15 (只读)	[127:64]	esw_pktin_cnt2	进入 GOE 模块的分组计数器
		[63:0]	esw_pktin_cnt3	GOE 模块往 0 口输出的分组计数器
EOS	16	[127:64]	eos_mdin_cnt0	0 号端口对应的 EOS 模块进入的 TSN_MD 计数器。
		[63:0]	eos_mdout_cnt0	0 号端口对应的 EOS 模块传输的 TSN_MD 计数器。
	17	[127:64]	eos_mdin_cnt1	1 号端口对应的 EOS 模块进入的 TSN_MD 计数器。
		[63:0]	eos_mdout_cnt1	1 号端口对应的 EOS 模块传输的 TSN_MD 计数器。
	18	[127:64]	eos_mdin_cnt2	2 号端口对应的 EOS 模块进入的 TSN_MD 计数器。
		[63:0]	eos_mdout_cnt2	2 号端口对应的 EOS 模块传输的 TSN_MD 计数器。
	19	[127:64]	eos_mdin_cnt3	3 号端口对应的 EOS 模块进入的 TSN_MD 计数器。
		[63:0]	eos_mdout_cnt3	3 号端口对应的 EOS 模块传输的 TSN_MD 计数器。

附录 D　网络中支持的最大 TS 流量能力的分析

目前 BufM 中设计了 16 个（报文在 BufM 中缓存的位置）ID，一个 ID 对应 2Kb 的存储空间，足够缓存一个最长分组，可知 BufM 中最多能同时存储 16 个报文，一个报文对应一个 TS_metadata。在保证 TS 分组不出现丢包的情况下，根据乒乓队列的读写特点，当在一个时间槽内有 16 个 TS 分组写入 BufM，即有 16 个相应的 TS_metadata 写入队列（若为奇数时间槽，则 16 个 TS_metadata 全写入 Q1 队列；若为偶数时间槽，则 16 个 TS_metadata 全写入 Q0 队列）时，网络中的 TS 流量达到最大。在网络中以太网最小帧间距 min_frame_gap 是 12 字节，以太帧前导码 preamble 为 7 字节、帧开始符 sfd 为 1 字节，队列长度为 queue_length，报文长度为 pkt_len（单位字节；以太网报文长度最小为 64B，最大为 1518B），时间敏感流量乒乓队列切换的时间槽为 time_slot（单位 us），网络中支持的最大 TS 流量为 ts_rate（单位 bps），则可得如下公式：

$$\frac{(\text{time\_slot} * 10^{-6}) * \text{ts\_rate}}{8} = \text{queue\_length} * (\text{pkt\_len} + \text{min\_frame\_gap} + \text{preamble} + \text{sfd})$$

可得 ts_rate 的计算公式

$$\text{ts\_rate} = \frac{\text{queue\_length} * (\text{pkt\_len} + \text{min\_frame\_gap} + \text{preamble} + \text{sfd}) * 8}{(\text{time\_slot} * 10^{-6})} = \frac{16 * (\text{pkt\_len} + 20) * 8}{\text{time\_slot} * 10^{-6}} = \frac{128 * (\text{pkt\_len} + 20) * 10^{6}}{\text{time\_slot}}$$