TSNSwitch Hardware Design Documentation

(V1.0)

OpenTSN TSNSwitch

content

1. Introduction to the TSNSwitch project

This document is the hardware design document of the sub-project TSNSwitch in the OpenTSN open source project, including several main parts of TSNSwitch design goals and module design.

1.1 Design goals of TSNSwitch

TSNSwitch is a sub-project under the OpenTSN project, and its design goal is to realize the development of a switch-related function that satisfies TSN network transmission. The specific functions are as follows:

1. Use 1588 ptp protocol for time synchronization of the entire network; 2. Output scheduling mechanism based on CQF; 3. Traffic reservation scheduling based on P3-P5 priority based on token bucket; 4. Support local TSN nodes to pass local status Periodically report beacon messages; 5. Provide TAP port for mirroring for analysis; 6. Support ring topology, star topology, linear topology and other topologies.

1.2 Design scheme of TSNSwitch

1.2.1 Using the platform

TSNSwitch is designed and implemented based on the FAST architecture. The FAST architecture is shown in Figure 1 below. For details, please refer to FAST Design document "FAST Open Source Platform Principles and Applications":

Figure 1 FAST Architecture Diagram

The entire logic development of TSNSwitch is carried out on openbox, which supports FAST architecture design, The appearance of openbox is shown in Figure 2 below. Openbox has a total of 4 1Gbps data network ports 0-3.

Figure 2 Appearance of Openbox

1.2.2 Requirements

Analysis Typical topology scenarios of TSN networks include linear, ring, star, tree, etc.

Two sets of architectures are designed: a ring architecture and a non-ring architecture.

Ring structure: Ports 0 and 1 are used as the internal connection ports of the TSN network;

system, etc.; port 3 is used as TAP port for data mirroring for analysis.

Figure 3 below is a ring topology, the data network ports 0 and 1 of each node support deterministic forwarding;

According to the data, it can enter through No. 0 and exit from No. 1, or can enter from No. 1 and exit from No. 0. It needs to pass TSNSTM (TSN control system).

The software responsible for static configuration in the controller) is configured.

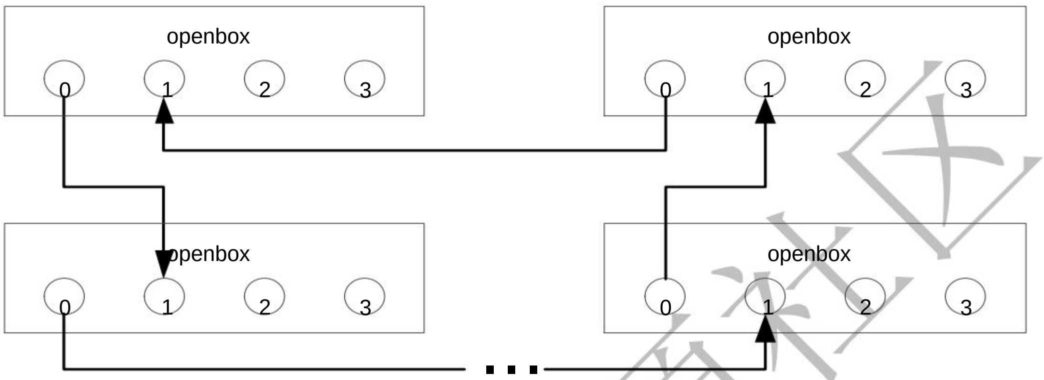

Figure 3 Ring topology

Non-ring-shaped architecture: Ports 0, 1, 2 and 3 can be used as the internal connection ports of the TSN network, and also

It can be used as a device port for external terminal systems, etc.; at the same time, it provides TAP function selection, users can configure 3

The TAP function of the port is used to collect the mirror image of the output traffic of the ports 0, 1 and 2 of the switch for analysis. need attention

Yes, when using this scheme to build a topology, it is not allowed to have a ring structure in the network, because when there is a ring,

The ptp synchronization sync (broadcast) message will continue to circulate in the ring, resulting in slave time within one synchronization cycle of the master clock

The clock node receives multiple synchronization sync messages, which cannot be processed by the hardware logic.

Figure 4 below is a star topology, and the four data network ports of each node need the deterministic forwarding function;

The flow direction is planned by TSNSTM, and there are many different routes for data flow transmission.

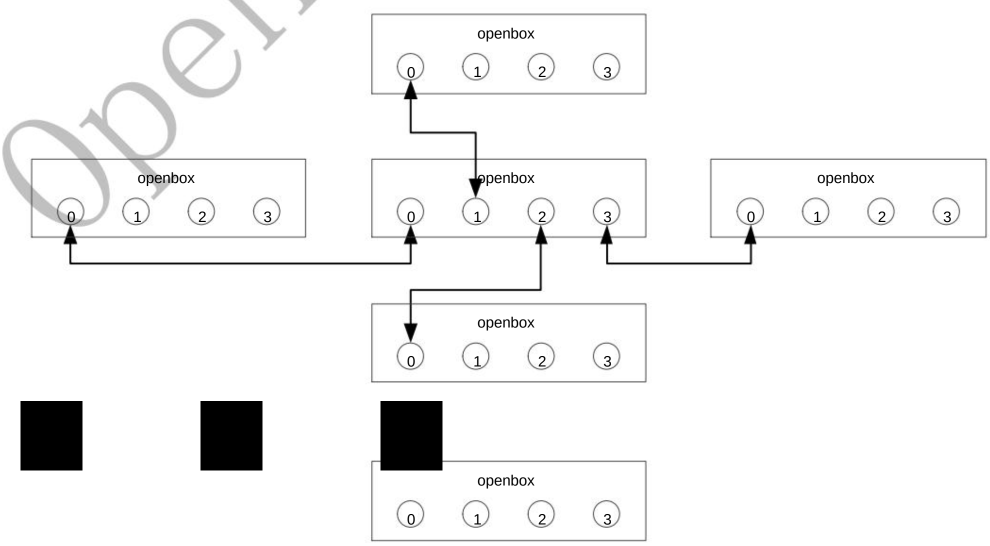

Figure 4 Non-ring topology

2. Overall design of TSNSwitch hardware

TSNSwitch provides two sets of overall architectures, corresponding to ring topology and non-ring topology respectively.

An introduction.

2.1 Ring topology architecture design

As described in Section 1.2.1, TSNSwitch is designed with the FATS architecture, but for TSNSwitch's

This simplifies the design of the control path. The architecture of TSNSwitch is shown in Figure 5 below. The white module is TSNSwitch.

The key modules that need to be added need to be redeveloped, and the orange modules need to be partially revised to support the TSNSwitch function.

Modified module.

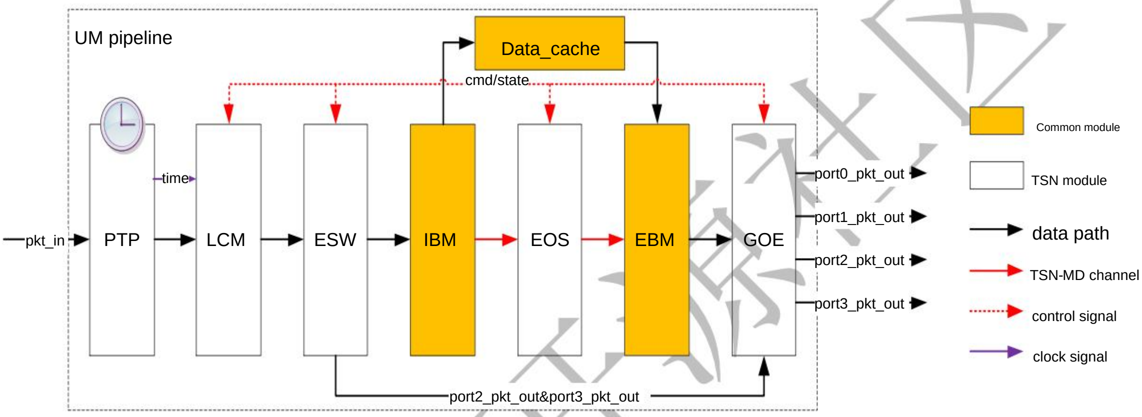

Figure 5 Overall architecture design of TSNSwitch in ring topology

The key modules include the following four parts:

PTP (Precise Time Protocol) module: It is the time synchronization module of TSNSwitch, using 1588ptp

The protocol is used to synchronize the time of the whole network. For the specific design, refer to the document "TSNSwitch Clock Synchronization Design Document".

LCM (Local Control Management) module: It is the local management and control module of the TSN switching node.

yuan. Mainly responsible for maintaining the maintenance and request of each module control related registers, periodic equipment status information reporting, processing

Handle register modification requests from TSNSTM nodes.

ESW module (Epipe SWitching): It is the switching module of the TSN switching node. Mainly responsible for grouping type

Parsing, traffic policing, forwarding action generation, and TS_metadata (TS_MD, a metadata specific to TSN switches

data) generated operations. This module divides the packets entering the pipeline into four categories: control related (ptp) packets, TS

Grouping, reserved bandwidth (RC) grouping, and best effort (BE) grouping. BufM resource tightness in the pipeline can

In order to cause packet loss, discard low-priority packets according to the above priorities to ensure reliable transmission of TS packets. in addition,

ESW also needs to determine the destination port of all packets according to the configuration of TSNSTM.

EOS (Epipe Output Scheduling) module: It is the output queue scheduling module of the TSN node, which is used to implement the

Now simplified cqf forwarding. Mainly responsible for supporting the circular queue forwarding function of the core TS_metadata to ensure that TS

Deterministic delay of packet scheduling output, and implementation of token bucket-based traffic shaping function for reserved bandwidth packets.

GOE (General Output Engine) module: mainly responsible for receiving packets from EBM according to

The "output" field in FAST_metadata is sent with packets from the ESW, and sent to all sent messages.

Statistics are counted.

Processing flow: The first-level module is the PTP module. The PTP module is based on the protocol field of the packet and the purpose of the packet. MAC to distinguish packets, if the packet needs to be processed, it will be processed in the internal logic of the PTP module, and other packets will be processed next to it. road off. The second-level module is the LCM module. LCM is distinguished according to the protocol field of the packet and the destination MAC of the packet. Grouping, extract the information in the beacon configuration message and fill it in the corresponding register, other messages are bypassed Lose. The third-level module is the ESW module. The ESW module is based on the packet source MAC, input port number, destination The MAC is used to distinguish the packets, and the packets are forwarded to the lower-level module, forwarded to the GOE module, and discarded. the first The fourth-level module is the IBM module. The IBM module is responsible for writing the packet data into the RAM of the data_cache module. And get the stored ID number from data_cache, store the ID in TS_metadata and transmit it to lower-level modules. the first The fifth-level module is the EOS module. EOS divides TS_metadata into 3 types according to the field information in TS_metadata type and write it into 4 kinds of queues according to the current time slice information, and then use the CQF queue forwarding mechanism to TS_metadata for scheduling. The sixth-level module is the EBM module, and the EBM module is given according to the EOS module. The ID number in TS_metadata is used to read packet data to RAM. The seventh-level module is the GOE module, which will be divided into Groups are differentiated and transmitted based on output ports. Data_cache module, which uses RAM for cache management of packets, And use fifo to manage the address of the RAM block. After the group is cached in the RAM, the stored ID number is obtained. Then according to the ID number to the corresponding RAM address to read in groups.

2.2 Non-ring topology architecture design

In order to support a variety of TSN network topologies, it is necessary to design a bidirectional authentication system that is different from the ring structure. TSNSwitch for the scheduled transmission.

For TSN switching nodes, in order to support diverse topologies, all four network ports need to be used for TSN In the network, the original TAP port function needs to be retained at the same time. All 4 network interfaces feature deterministic forwarding and can be used for Connections within the TSN network can also be connected to external devices. Considering the demand for multicast packets in actual scenarios, add Add forwarding processing logic for multicast packets.

The overall architecture design of TSNSwitch in a non-ring topology is shown in Figure 6 below. The blue modules are for TSNSwitch Functions to support modules that need to be partially modified.

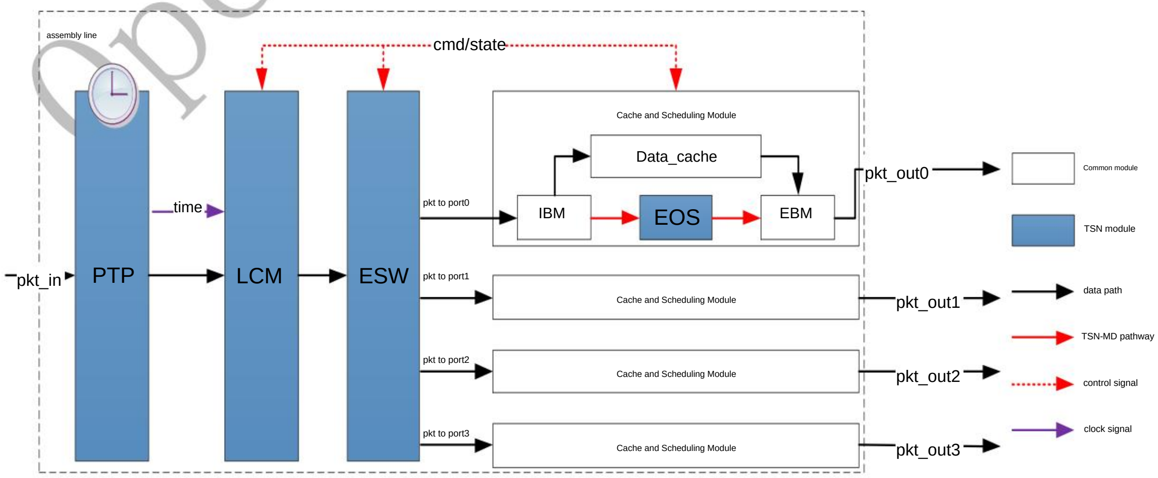

Figure 6 Overall architecture design of TSNSwitch in non-ring topology

The key modules include the following five parts:

PTP module: Consistent with the architecture design of the ring topology.

LCM module: Consistent with the architecture design of the ring topology.

ESW module: It is the switching module of the TSN switching node. Mainly responsible for packet type analysis, traffic monitoring, TS_metadata Generates, looks up the forwarding table, and transfers the packet to the cache and scheduling module corresponding to the port. This module divides the packets entering the pipeline into four categories: control-related packets, TS packets, reserved bandwidth packets, and best effort packets. When the shortage of BufM resources in the pipeline can lead to packet loss, the low-priority packets are discarded according to the above priorities to ensure reliable transmission of TS packets.

Cache and management module: This module includes IBM (Input Buffer Management), data_cache, EOS, EBM (Extract Buff Management) module. There are a total of 4 cache and management modules, each module corresponds to a port, the grouped data is cached in the data_cache, and the EOS module schedules and manages according to the grouped TS_metadata.

EOS module: Consistent with the architecture design of the ring topology.

Processing flow: The first-level module is the PTP module. The PTP module distinguishes the packet according to the protocol field of the packet and the destination MAC of the packet. If the packet needs to be processed, it is processed in the internal logic of the PTP module, and other packets are bypassed. The second-level module is the LCM module. The LCM distinguishes the packet according to the protocol field of the packet and the destination MAC of the packet, extracts the information in the beacon configuration packet and fills it in the corresponding register, and bypasses other packets. The third-level module is the ESW module. The ESW module searches the forwarding table according to the grouping destination MAC and input port number, and transmits it to the corresponding lower-level module according to the output port number obtained after the search. The fourth-level module is the IBM module. The IBM module is responsible for writing the packet data into the RAM of the data_cache module, and obtains the stored ID number from the data_cache, stores the ID in TS_metadata and transmits it to the lower-level module. The fifth-level module is the EOS module. EOS divides TS_metadata into 3 types according to the field information in TS_metadata and writes them into 4 types of queues according to the current time slice information, and then uses the CQF queue forwarding mechanism to schedule TS_metadata. The sixth-level module is the EBM module. The EBM module reads the packet data to the RAM according to the ID number in the TS_metadata given by the EOS module. The seventh-level module is the GOE module, which differentiates and transmits the packets according to the output ports. The Data_cache module uses RAM to manage the group's cache, and uses fifo to manage the address of the RAM block. After the group is cached in RAM, the stored ID number is obtained, and then the corresponding RAM address is read according to the ID number.

2.3 Differences in the design of the two topology architectures

2.3.1 Differences in overall topology The

architecture design of ring topology only supports the topology where all nodes form a ring; the architecture design of non-ring topology All topologies are supported except those containing rings, such as star, linear, etc.

2.3.2 The difference between ports

In the architecture design of the ring topology, only ports 0 and 1 have the deterministic forwarding function, while port 2 is fixed. As the port of the external device, port 3 is fixed as the TAP port.

The four ports in the non-ring topology design have deterministic forwarding functions, and all four ports can be used as ports for external devices without specific constraints. a function.

2.3.3 Differences in UM Internal Processing

The internal processing of the two topology architectures is mainly on the ESW module. In the ring topology architecture, the ESW module only divides the packets into two types: packets sent to the external devices of the node and packets sent to the TSN ring. In the non-ring topology structure, the ESW module needs to look up the table and forwarding table for the packet, and transmit it to the corresponding subordinate module according to the output port number of the forwarding table.

3. TSNSwitch module design

3.1 LCM module

3.1.1 LCM module function

According to the description in the previous section, the LCM module is responsible for 1) periodically reading the values of all local control-related registers, and constructing an Ethernet frame containing the beacon report message for transmission; 2) parsing the received Ethernet frame of the beacon update message from the TSNSTM node network frame, and modify the corresponding register according to the corresponding field value. 3) Discard the beacon packets sent by the local LCM and returned to the local LCM when the destination address is not found.

3.1.2 LCM Module Design In order

to simplify the design, the control path of the original FAST architecture is deleted in the implementation of TSNSwitch, and the LCM module (local management module) is used to obtain and update the global status information of TSN nodes. The LCM accepts the management control of the remote controller (TSNSTM) through beacon messages. The LCM is responsible for the implementation of the following two functions:

1. The LCM module periodically constructs a frame (report message) containing the beacon message, and fills in the status information of the node into the beacon report message. 2. After receiving the register configuration message (update message) from the TSNSTM node, the readable and writable fields will be

The value is read out and replaces the original value in the corresponding register.

3. Detect the beacon packets entering the LCM module. If the beacon packets are sent by the local LCM (which can be judged according to the SMAC address), the packets will be discarded.

The overall architecture of the LCM module is shown in Figure 7:

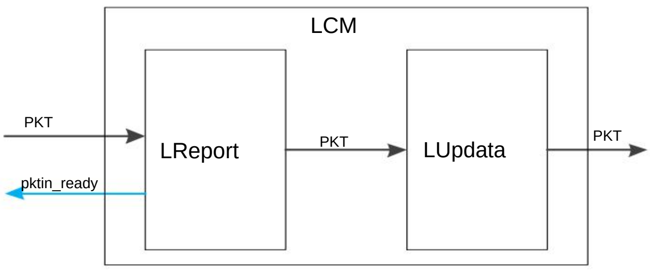

Figure 6 Overall architecture of LCM module

LReport (LCM Report) module: responsible for receiving and sending frames from the PTP module to the LUpdate module (sent from the LUpdate module to the ESW module); periodically reporting a beacon according to the global clock

Message (the period can be customized, the tentative period in the current scheme is 32ms), when the clock reaches the reservation period, LReport back-pressures the PTP module, constructs a Beacon Report message containing the current node status information, and sends it into the LUpdate module.

LUpdate (LCM Updata) module: After receiving the Beacon Update message frame from the LReport sub-module, it is responsible for judging whether the message frame is sent by the local LCM. The value of , updates the corresponding register. If the received frame is not a Beacon Update message frame, it will be forwarded directly to the ESW.

3.2 ESW module

3.2.1 ESW module function

The main functions of the ESW module are as follows:

1) Packet type analysis; 2)

MAC address comparison; 3)

Packet forwarding; 4)

Modification of FAST_metadata data and construction of TS_MD (TS_metadata) data; 5) Traffic monitoring.

3.2.2 Design of ESW module

Since the ESW module is very different between the ring topology and the non-ring topology, the following two parts will be introduced.

3.2.2.1 Ring topology

The main functional requirements of the ESW module are: packet type parsing, MAC address comparison, FAST_metadata modification of TS_MD data, packet forwarding, and traffic policing.

After the packet enters the ESW module, it needs to parse the packet, and transmit the packet type to the EOS module for the EOS module to enter different queues according to different types of packets; if the purpose of the packet is to directly connect the device locally, it will be sent directly by the ESW module. Count to the GOE module; other packets are transmitted to the lower-level modules. Packets transmitted to lower-level modules also need to be discarded according to the number of bufm idle IDs in the pipeline (only non-TS packets are discarded).

To meet the requirements of TAP, copy all the packets that need to be forwarded to the TAP port for forwarding.

In the design, the ESW module is designed as three modules: ESW top-level module, PKE module, and PAC module. Tool

The body frame diagram and sub-modules are implemented as follows:

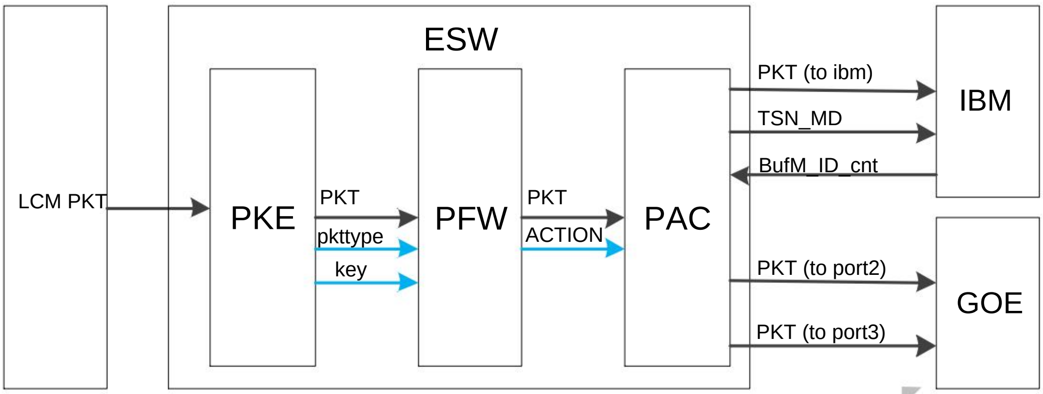

Figure 7 ESW module division diagram of ring topology

PKE (Pkt **Key** Extract) module: determine whether it is a packet with a VLAN header or a standard Ethernet packet. if For packets with VLAN headers, the packet types are distinguished according to the PCP value: TS packets, reserved bandwidth packets, and best effort packets; for standard Ethernet packets, the packet types are distinguished according to the protocol field: ptp packets, best effort packets grouping. And write the result of parsing into pkttype; at the same time, extract the triplet of the packet {destination MAC, source MAC, Input port} is transmitted to the PFW module at the same time as the packet.

PFW (Pkt **Find** ForWarding) module: compares the information of the triplet with the MAC of the local directly connected device Compare. Source MAC comparison: according to the source MAC comparison, determine whether the packet is to be discarded or continue to the destination MAC Compare the two treatments. Destination MAC comparison: the destination MAC is all F, the destination MAC is equal to the MAC of the local directly connected device, The destination MAC is not equal to the MAC of the local directly connected device. Construct ACTION information based on the comparison result.

PAC (Pkt ACtion) module: forwards packets according to the content in ACTION. Unicast: If ACTION If the output port is 2, the grouped FAST_metadata data is modified according to ACTION and then grouped to this If the grouping output port of ACTION is not 2, it will be grouped according to the idle ID of bufm discarded or forwarded. When bufm is greater than 3 (indicating that bufm has at least 4 free blocks for buffering packets), the grouping All can directly modify the FAST_metadata and then transmit the packet; when bufm is equal to 3, the packet type of best effort needs to be discarded, and other packets are transmitted normally; when bufm is greater than 0 and less than 3, the packet type For reserved bandwidth or best effort, it needs to be discarded, and other packets are transmitted normally; when bufm is 0, All packets are discarded. When not discarded, TS_MD needs to be constructed during packet transmission. When broadcasting: processing and single In addition to the same processing method as broadcast, a copy of the packet needs to be forwarded to port 2 for forwarding. forwarding at all All packets are copied and forwarded to the TAP port.

3.2.2.2 Non-ring topology

The non-ring topology architecture is the same as the ring topology architecture in the overall module design, but the functions of each module are slightly different. At the same time, the direction of the message output is also different, which will be described in detail below.

The ESW module is designed into 3 modules like the ring topology design: PKE module, PFW module, PAC module. The specific frame diagram and sub-module implementation are as follows:

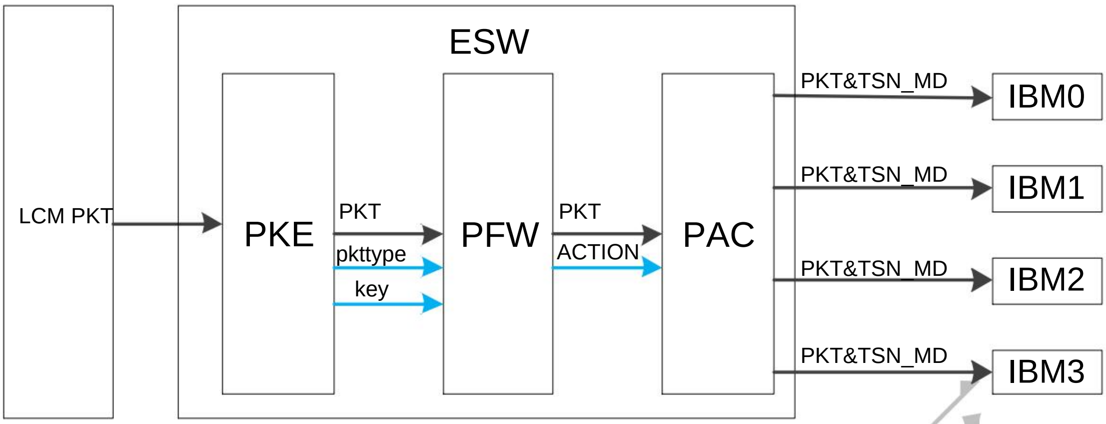

Figure 8 ESW module division diagram of non-ring topology architecture

In this architecture, the key data of the ring topology architecture is basically the same, only the key field is slightly different.

PKE module: determine whether it is a packet with a VLAN header or a standard Ethernet packet. If it is a packet with a VLAN header, then Distinguish packet types according to PCP value: TS packet, reserved bandwidth packet, best effort packet; if standard Ethernet packets are distinguished by the packet type according to the protocol field: ptp packets and best effort packets. and will parse The result is written into pkttype; at the same time, the two-tuple {destination MAC, input port} of the packet is extracted and the packet is transmitted at the same time to the PFW module.

PFW module: distinguish whether the packet is a broadcast, multicast or unicast packet according to the destination MAC address information of the packet; if If it is a unicast or multicast packet, look up the forwarding table, obtain the output port according to the result of the table lookup, process the message, and forward it. Send or discard; if it is a broadcast message, the output port of the message will be sent to the three ports except the input port. According to the packet output type, the packet type information obtained when the PKE module parses the packet, and the output port, there are three total Information constructs ACTION. In addition, when the user selects the TAP function, all incoming packets from port 3 need to be discarded. Copy the packets transmitted by ports 0, 1, and 2 of the switching node to port 3 for output.

PAC module: This module is responsible for traffic supervision, construction of TS_MD, and selection of TAP mode, while grouping root It is distinguished according to the output port and sent to the corresponding IBM module. When bufm is greater than 3, grouping can be performed directly The packet is transmitted after FAST_metadata is modified; when bufm is equal to 3, the packet type is best effort needs to be processed The line is discarded, and other packets are transmitted normally; when bufm is greater than 0 and less than 3, the packet type is reserved bandwidth or The best effort needs to be discarded, and other packets are transmitted normally; when bufm is 0, all packets are sent. line discarded. When not discarded, TS_MD needs to be constructed during packet transmission.

3.3 EOS module

3.3.1 EOS module function

In order to support FAST-based TSN exchange, it needs to be implemented in the EOS module:

1. Cache TS_metadata data of different types of packets in queues, that is, use Q0 and Q1 to cache TSN packets The TS_metadata data of the message, use the Q2 cache ptp and the TS_metadata data of the reserved bandwidth message, And use Q3 to cache the TS_metadata data of best effort Ethernet packets;
2. Schedule the queue according to the message priority, Q0 = Q1 > Q2 > Q3;

3. Implement CQF scheduling function, that is, use Q0 and Q1 to implement ping-pong queue;

4. Shape the reserved bandwidth traffic (that is, the TS_metadata data of queue Q2) based on the token bucket mechanism;

5. Configure the token bucket parameters, which currently include two, namely the bucket depth of the token bucket and the speed of adding tokens to the token bucket.

Rate;

6. For other modules (LCM) to read EOS module related register information.

3.3.2 EOS Module Design

The main function of the EOS module is to implement the CQF scheduling mechanism, and at the same time perform the RC grouping based on the token bucket mechanism. If the RC packet bandwidth exceeds the reserved bandwidth, the RC packet discard signal will be sent to the lower-level module EBM (in the The EBM module discards the RC packet).

There are three important parameters in the module: the parity of the time slice, the bucket depth of the token bucket and the rate of injecting tokens into the token bucket. The number needs to be configured by LCM.

This section mainly introduces the overall architecture design of the EOS module. The overall architecture of the EOS module is shown in Figure 10:

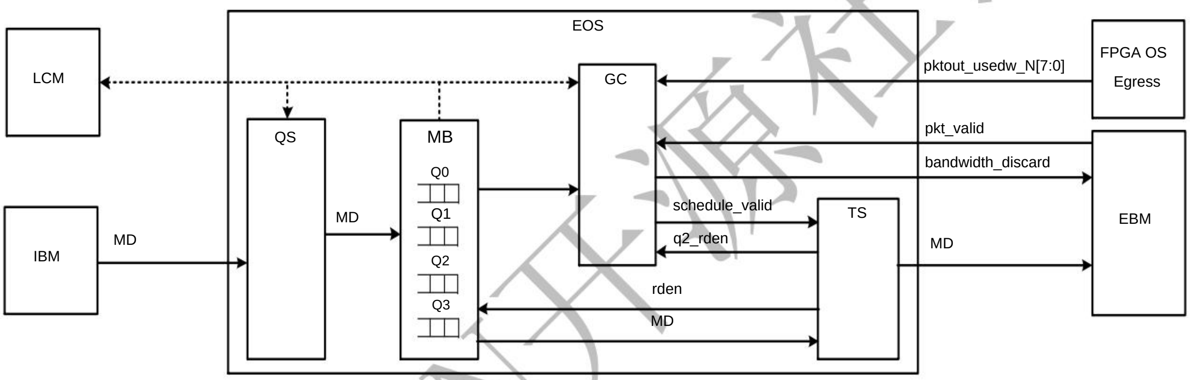

Figure 9 The overall architecture of the EOS module

QS (Queue Selecting) module: Queue selection module, which writes TS_metadata data into the corresponding queue. According to the parity of the current time slot and TS_metadata data [23:21] (indicates the grouping class corresponding to the TS_metadata type field) determines which queue to transfer TS_metadata data to: if TS_metadata data[23:21] is 3 (table indicates that the grouping type corresponding to this TS_metadata is TS grouping), and the current time slot is an odd number, then the TS_metadata Data[8:0] (the packet output port number and the location ID buffered in BufM) are transferred to Q1; if TS_metadata Data[23:21] is 3, and the current time slot is even, then transfer TS_metadata data[8:0] to Q0; if TS_metadata data [23:21] is 2 (indicating that the grouping type corresponding to the TS_metadata is ptp grouping), then the TS_metadata data [19:9] (packet length field) is set to 0 (ptp packets do not need to be consumption token), if the TS_metadata data [23:21] is 1 (indicating that the grouping type corresponding to the TS_metadata is RC grouping), then subtract two beats of FAST_metadata length (32 bytes) from TS_metadata data [19:9], and connect The same as TS_metadata data [8:0] is transmitted to Q2; if TS_metadata data [23:21] is 0 (indicating that the The packet type corresponding to TS_metadata is BE packet), then transmit the TS_metadata data [8:0] to Q3.

MB (Metadata Buffer) module: TS_metadata data buffer module, Q0 is for buffering TS packets Even-numbered clock queue for TS_metadata data, Q1 is the odd-numbered clock for buffering TS_metadata data of TS packets Queue, Q2 is the TS_metadata data queue for buffering RC packets and PTP packets, Q3 is the data queue for buffering BE packets

TS_metadata data queue. Wait for the TS_metadata data read signal of the TS module, and transfer the corresponding queue

TS_metadata data read out.

GC (Gate Control) module: gate control module. According to the parity of the current time slice, whether the Q0 and Q1 queues are empty,

Whether the port fifo has enough free space to determine whether the Q0 and Q1 queues can be scheduled;

is empty, whether the port fifo has enough free space to determine whether the Q2 queue can be scheduled; according to the Q3 queue is

If it is not empty, whether the port fifo has enough free space to judge whether the Q3 queue can be scheduled;

The judgment result of whether the Q2 and Q3 queues can be scheduled is sent to the TS module. At the same time, based on the token bucket mechanism, the RC flow is processed.

Shaping to decide whether the RC packet should be discarded in the subordinate module EBM.

TS (Transmiting and Scheduling) module: scheduling sending module, according to the Q0, Q1,

The judgment result of whether the Q2 and Q3 queues can be scheduled, and the absolute priority scheduling strategy is used for the queues that can be scheduled.

Schedule TS_metadata for queues in MB modules.

3.4 GOE module

According to the analysis of the GOE module, the design of the GOE module is as follows:

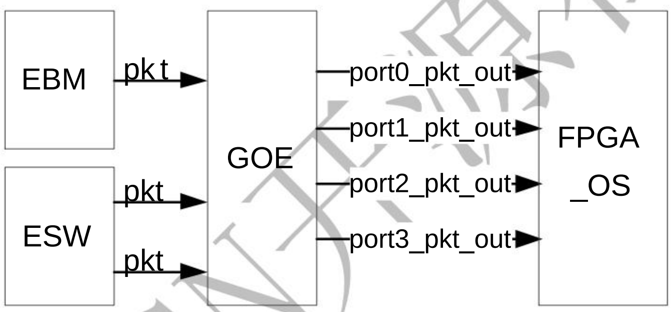

Figure 10 GOE module structure diagram

The function of this module is relatively simple. It determines which interface to transmit according to the incoming packets, and then groups the packets.

transmitted to the corresponding interface. All transmitted packets are counted simultaneously.

3.5 Switch Management Control

The TSN network is uniformly configured and managed by TSNSTM. TSNSTM implements ptp messages for each node.

For management, the ptp message specifies the destination node that needs to be configured through the destination MAC. Therefore, each node needs to have its own

One of its own MAC addresses, this design specifies that the first 40 bits of the TSN network node MAC are 00:06:06:

00:00, the last 8 bits can be configured by software to distinguish different nodes.

After TSNSTM knows the MAC address of each node, it can send a ptp message for configuration.

After the ptp message whose destination MAC is the local MAC, some specific field information in the message will be extracted to the local sender.

The value of the register is updated.

Each node will also periodically report a ptp message to TSNSTM, which carries some information about the node.

Including BufM usage, counters for each module, etc. The user can pass the received ptp message on the TSNSTM

information to understand the running status of each node, if a node or multiple nodes do not report the message can be directly in the

TSNSTM can locate which switch the problem is on.