

OpenTSN3.4 组网演示系统

使用手册

(版本 1.0)

OpenTSN

OpenTSN 开源项目组

2022 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	2022.5	初版编制	开源项目组	OpenTSN3.4

目录

1、概述3

2、组网环境组成3

 2.1 TSN 组网环境拓扑.....3

 2.2 组网设备配置.....3

 2.3 硬件板卡说明.....4

 2.4 软件使用说明.....5

3、硬件逻辑代码固化.....6

4、演示案例.....7

 4.1 时间同步精度测试7

 4.1.1 预期结果.....7

 4.1.2 参数配置.....8

 4.1.3 操作步骤.....8

 4.1.4 测试结果.....9

附录 1: 软件运行环境搭建10

附录 2: TTE 组网拓扑.....17

1、概述

本文通过搭建组网案例来对 opentsn3.4 版本 TSN 网卡和 TSN 交换机逻辑的功能准确性进行演示验证，以 802.1AS 时间同步为例。

2、组网环境组成

2.1 TSN 组网环境拓扑

使用 2 个 TSN 交换机、4 个 TSN 网卡、2 台 PC（PC 机之间确保能够正常 ping 通）搭建如图 2-1 所示的演示环境拓扑（GM 为主时钟节点、bc 为边界时钟节点、slave 为从时钟节点），图中实线均为网线连接，箭头指向即为数据流向。

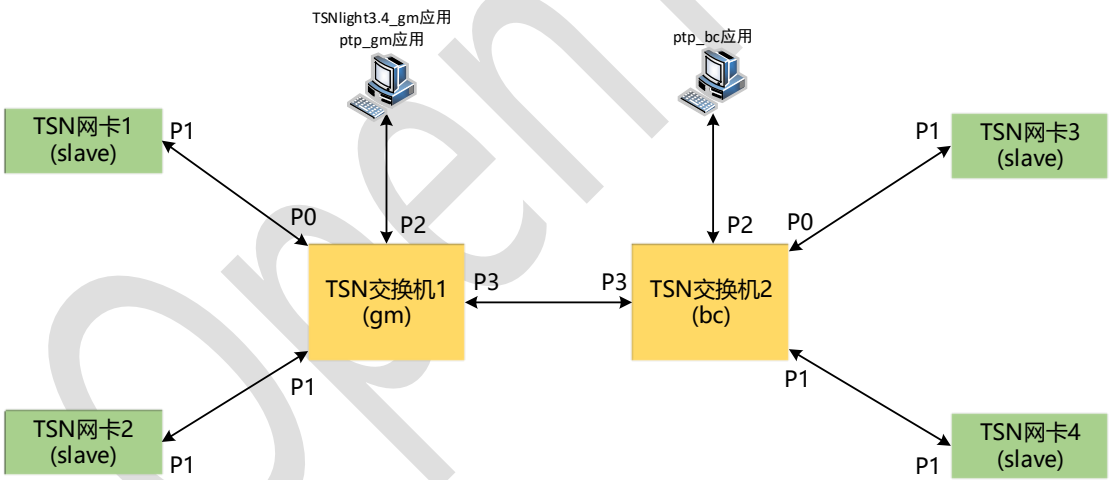


图 2-1 组网环境拓扑图(TSN)

2.2 组网设备配置

组网中各设备的信息如下表 2-1 所示。

表 2-1 设备配置

序号	设备名称	配置信息	备注
1	TSN 交换机	2 个，搭载了 OpenTSN3.4 的交	用于 TSN 网络中流量

序号	设备名称	配置信息	备注
		交换机逻辑	的交换
2	TSN 网卡	4 个，搭载了 OpenTSN3.4 的网卡逻辑	用于 TSN 网络中流量的映射、重映射、注入等
3	TSNLight3.4_gm/ptp_gm 应用控制器	1) Linux 设备（虚拟机也可，具体要求参考附录） 2) 直连在 TSN 交换机 1 的 P2 口；	TSNLight3.4_gm 用于 TSN 网络配置、ptp_gm 管理网卡 1、2 时间同步等
4	ptp_bc 应用控制器	1) Linux 设备（虚拟机也可，具体要求参考附录） 2) 直连在 TSN 交换机 2 的 P2 口	ptp_bc 用于管理交换机 2、网卡 3、网卡 4 的时间同步等

2.3 硬件板卡说明

板卡的实物如下图 2-2 和图 2-3 所示，板卡尺寸为：100mm* 80mm。板卡内部根据固化的逻辑代码不同，实现的功能也不同。板卡可固化网卡、交换机硬件逻辑。

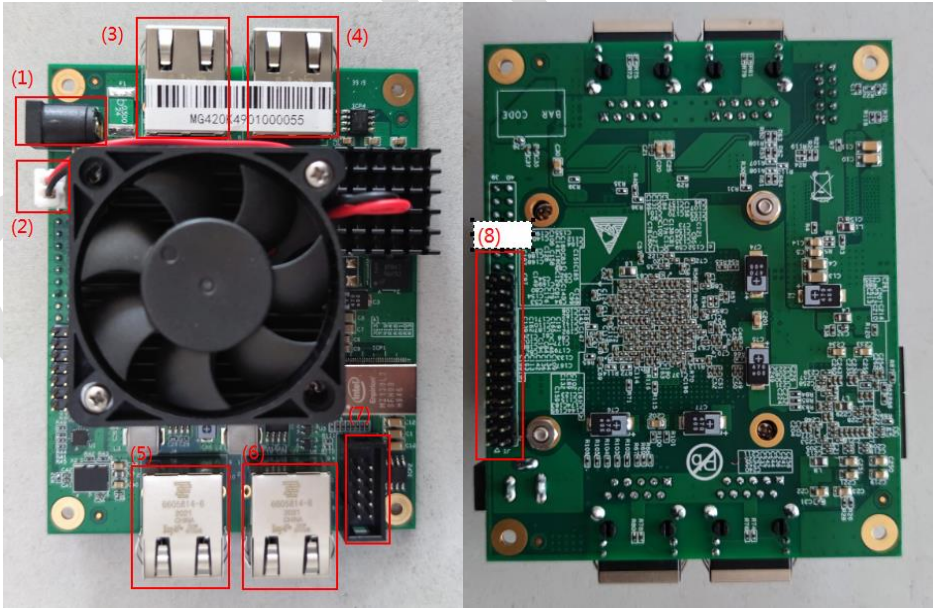


图 2-2 板卡-正面和背面



图 2-3 板卡-侧面

图 2-2 与图 2-3 中标注 (1)-(8)的接口，说明如表 2-2。

表 2-2 接口说明

编号	接口说明
1	12V 电源插座。注意此插座不能与底板上面的 12V 插座同时插上。
2	12V 风扇插座，有防呆设计。
3	千兆电口 3
4	千兆电口 2
5	千兆电口 0
6	千兆电口 1
7	JTAG 插座，有防呆设计。可用于 JTAG 边扫调试及 AS Flash 烧录。
8	2.54mm 网格插针

2.4 软件使用说明

针对提供的软件应用 TSNLight3.4_gm、ptp_gm、ptp_bc，编译以及使用方法都是相同的，下面以 ptp_gm 应用的编译及执行为例进行介绍。如图所示 2-4 所示，执行“make TYPE=TSN_FPGA”是对软件进行编译；执行“./ptp_app enp3s0”（enp3s0 网络接口名称）是软件启动；ptp_bc 应用的编译执行和 ptp_gm 应用保持一致；TSNLight3.4_gm 的执行命令是“./tsnlight enp3s0”。

```

root@ljs-E580: /home/ljs/桌面/opentsn3.4/ptp_gm
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
mid:3,offset=96
reference_pit = 0x5465579278,syn_clk_cor = 0x5465579218
sync timer out !
sync_seq 3267
mid:2,offset=64
reference_pit = 0x546b4d6fb0,syn_clk_cor = 0x546b4d6f70
mid:3,offset=96
reference_pit = 0x546b4d7890,syn_clk_cor = 0x546b4d7830
^C
root@ljs-E580: /home/ljs/桌面/opentsn3.4/ptp_gm# make TYPE=TSN_FPGA
gcc -g -o ./opensync/src/pkt_rec_api.o -c ./opensync/src/pkt_rec_api.c -lm -L .
./opensync/lib/ -loopensync -L ./opensync/src/ -lsim -I /usr/local/include/libxml
2/ -lxml2 -lpcap -lnet -I ./opensync/include
gcc -g -o ./opensync/src/pkt_snd_api.o -c ./opensync/src/pkt_snd_api.c -lm -L .
./opensync/lib/ -loopensync -L ./opensync/src/ -lsim -I /usr/local/include/libxml
2/ -lxml2 -lpcap -lnet -I ./opensync/include
gcc -g -o ./opensync/src/shadow_clock.o -c ./opensync/src/shadow_clock.c
./opensync/lib/ -loopensync -L ./opensync/src/ -lsim -I /usr/local/include/libxml2/ -l
lpcap -lnet -I ./opensync/include
gcc -g -o ./opensync/src/clock_config.o -c ./opensync/src/clock_config.c
gcc -g -o ./opensync/src/header_generate.o -c ./opensync/src/header_generate.c
gcc -g -o ./opensync/src/header_parse.o -c ./opensync/src/header_parse.c
ar -rc ./opensync/lib/libopensync.a ./opensync/src/pkt_snd_api.o ./opensync/src/
pkt_rec_api.o ./opensync/src/shadow_clock.o ./opensync/src/tools.o ./opensync/sr
c/clock_config.o ./opensync/src/header_generate.o ./opensync/src/header_parse.o
gcc -g -o ptp_app main.c ./tsninsight/tsninsight.c ptp_tsninsight.c 8021AS_P2P.c
-lm -L ./opensync/lib/ -loopensync -L ./opensync/src/ -lsim -I /usr/local/inc
lude/libxml2/ -lxml2 -lpcap -lnet -I ./opensync/include
root@ljs-E580: /home/ljs/桌面/opentsn3.4/ptp_gm# ./ptp_app enp3s0
*****device_info*****
dev_type 0
dev mid 6
*****gm_info*****
gm mid 0
sync_period 100000000
next_class_link mid 1
next_class_link mid 2
next_class_link mid 3
*****slave_info*****
slave mid 2
link_delay 840
*****slave_info*****
slave mid 3
link_delay 840
LIBNET INIT SUCCESS.
LIBPCAP INIT SUCCESS.
ser_addr.sin_addr.s_addr is c0a80179.

```

图 2-4 控制器使用截图示例

3、硬件逻辑代码固化

TSN 工程固化逻辑代码的具体操作步骤如下：

- 1) 建立好工程并编译完成后，生产固化文件，例如图 3-1。
- 2) 点击 tools->programmer->add files，添加编译完成的 .jic 文件（固化 TSN 网卡逻辑 tsnnic.jic 文件，固化 TSN 交换机逻辑选择 tsnswitch.jic 文件）。

3) 选择烧录线的 USB 串口, 并选择 JTAG 模式烧录, 点击 start 开始烧录 TSN 工程。如下 3-2 所示。



 tsnnic.jic	2022/5/16 20:36	JIC 文件	65,537 KB
 tsnswitch.jic	2022/5/16 20:35	JIC 文件	65,537 KB

图 3-1 jic 文件

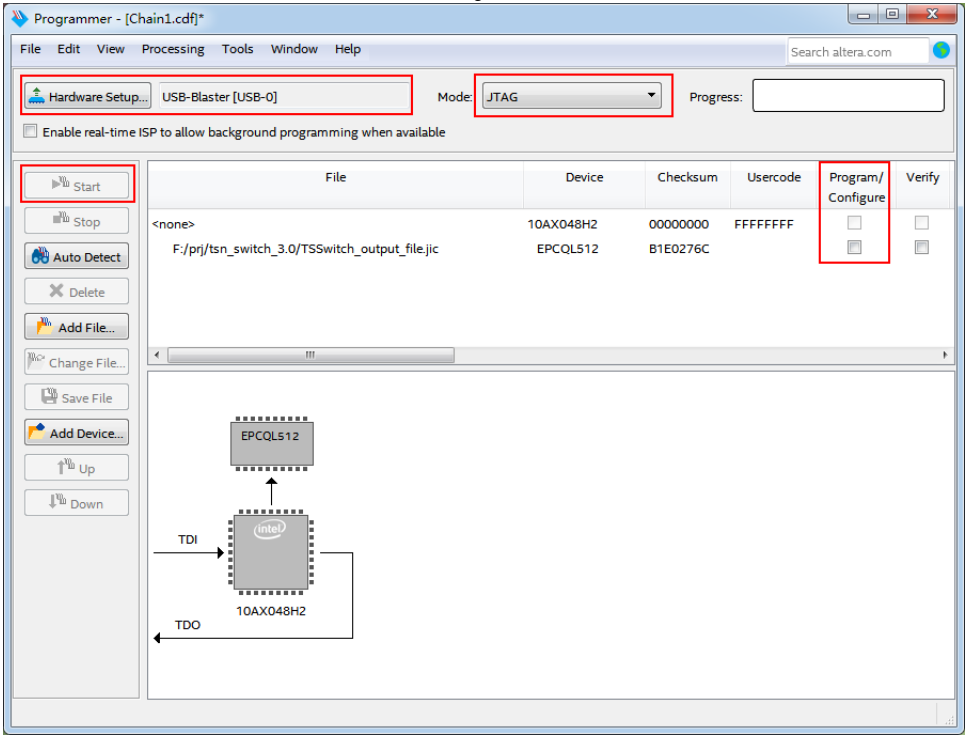


图 3-2 固化逻辑代码

4、演示案例

4.1 时间同步精度测试

4.1.1 预期结果

以 TSN 交换机 1 为主时钟（ground master）节点，TSN 交换机 2 为边界时钟（bounary clock）节点、TSN 网卡 1、TSN 网卡 2、TSN 网卡 3、TSN 网卡 4 作为从时钟（slave）节点的网络拓扑，进行时间同步精度测试，测试过程中记录并打印边界时钟节点及 4 个从节点的

offset 值, offset 值在 120ns 内。同时也可以利用示波器观测各个节点的波形脉冲, 脉冲信号 `cycle_start` 是每个调度周期开始时给出的脉冲使能, 脉冲信号 `cycle_start` 的时间差也应在 120ns 内。

4.1.2 参数配置

时间同步精度测试的关键表项配置在于时间同步报文的转发、链路延时(目前链路延时测量还在开发中, 目前配置的是静态链路延时, 相邻节点间估算值为 840ns) 与同步周期(100ms)的配置。

集中式控制器应用完成初始配置, PTP 应用发送 `sync` 报文转发给各个从时钟。组网的配置软件目前已配置好, 暂时无需关注, 连接好网络拓扑后, 按顺序运行软件应用, 就可以正常进行时间同步测试。

4.1.3 操作步骤

时间同步精度测试的具体操作步骤(软件启动要按顺序执行)如下:

- 1) 搭建好组网拓扑环境, 按照表 2-1 配置好终端设备环境;
- 2) 在 TSNLight3.4_gm 应用终端界面, 执行 “`./tsnlight enp3s0`” 运行集中控制器软件应用进行网络初始化配置;
- 3) 当看到界面打印进入 “`enter sync init_s`” 后, 在 `ptp_bc` 应用终端界面, 执行 “`./ptp_app enp0s17`” 运行 `ptp_bc` 应用, 使边界时钟节点进入等待接收 PTP 报文的状态;
- 4) 然后 `ptp_gm` 应用终端界面上, 执行 “`./ptp_app enp3s0`” 运行 `ptp_app` 软件应用进行同步报文下发;

- 5) 运行一段时间后，在 ptp_gm 软件应用界面，会打印编号 2（网卡 1）、3（网卡 2）的 offset 值；在 ptp_bc 软件应用界面，会打印编号 1（交换机 2）、4（网卡 3）、5（网卡 4）的 offset 值；
- 6) 同时提供脉冲信号测试接口（如图，从右往左数，第 3 个 I/O 针脚是脉冲信号线，第 10 个 I/O 针脚是地信号线），用于查看组网里各个节点间的脉冲波形。

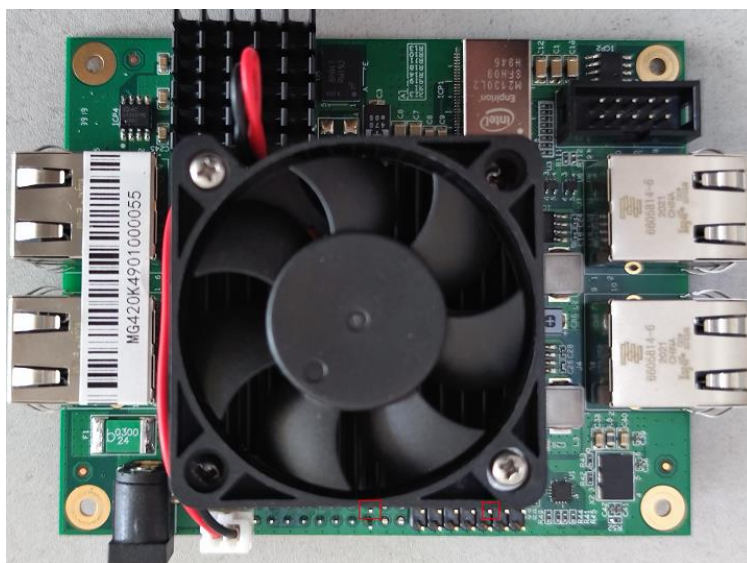


图 3-1 脉冲信号引脚接口

4.1.4 测试结果

以 100ms 时间同步周期进行网络各个节点的时钟对齐，打印界面的各个从时钟的 offset 值能够稳定在 120ns 之间变化，波形脉冲的时间差值在 120ns 以内波动。

因目前配置的链路延时只为估算值，没有具体测量，本文测试结果仅做参考。

附录 1: 软件运行环境搭建

软件需要使用 Linux 设备运行（建议使用 Ubuntu16.04LTS），并且需要安装 libnet、libpcap 以及 libxml 库。

本文提供两种方式供用户选择

方式一：使用 OpenTSN 提供的虚拟机运行集中控制器。

为了方便用户能够尽快搭建运行环境，减少因为搭建环境花费的时间，在百度网盘中提供已配置的虚拟机，下面具体介绍搭建虚拟机的步骤。

（1）下载百度网盘中的文件

从百度网盘中下载全部文件。

链接：https://pan.baidu.com/s/1Py6mpqs_nMmNU8frvkkuTA

提取码：85o9

（2）安装 VirtualBox-5.2.14-123301-Win

首选需要安装 VirtualBox 软件，安装后可以在该软件下导入需要安装的虚拟机。

（3）导入虚拟机

在 VirtualBox 中导入下载的虚拟机，首先在界面中点击“管理”，在管理中选择“导入虚拟电脑”，然后选择需要导入的文件（从百度网盘中下载的文件），然后选择“导入”。导入大约需要花费 5 分钟左右。

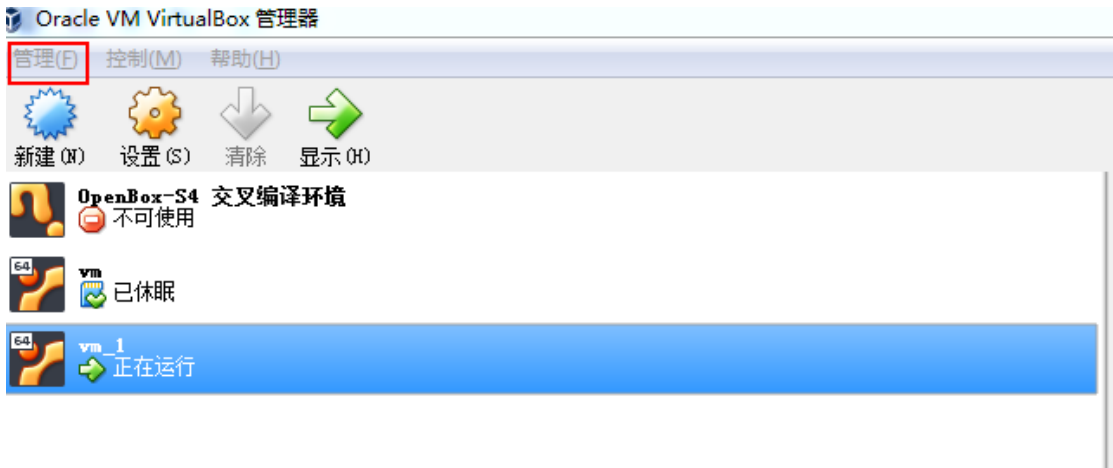


图 A-1 管理选项

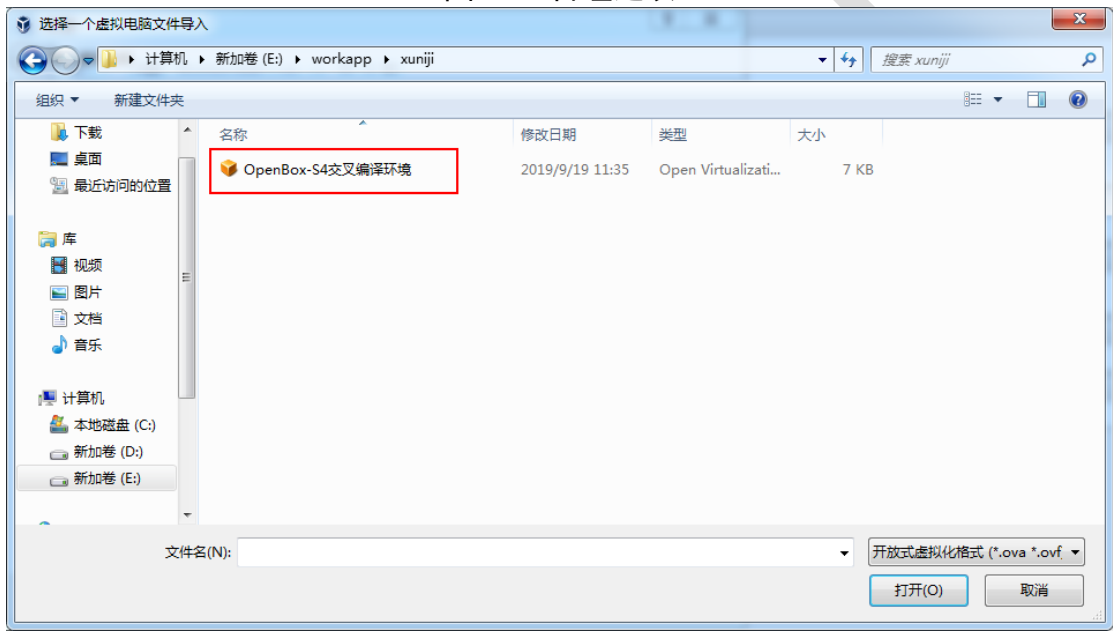


图 A-2 需要导入的文件

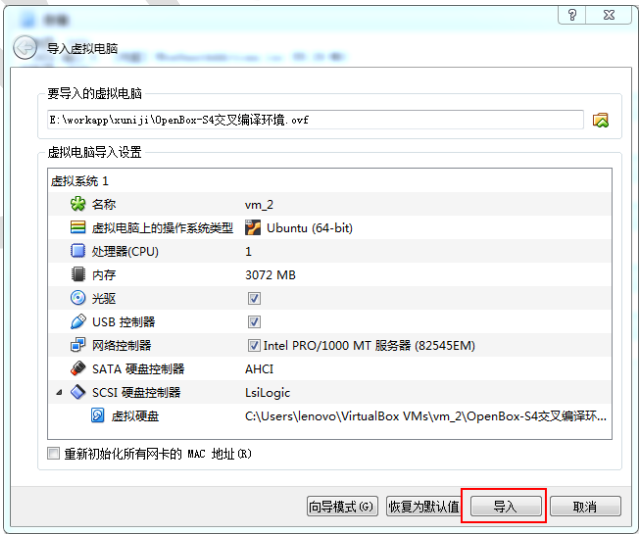


图 A-3 确认导入

(4) 虚拟机设置

导入成功后,需要对虚拟机进行设置。在 vbox 界面,点击“设置”,依次对存储、网络、共享文件夹进行设置。



图 A-4 虚拟机设置

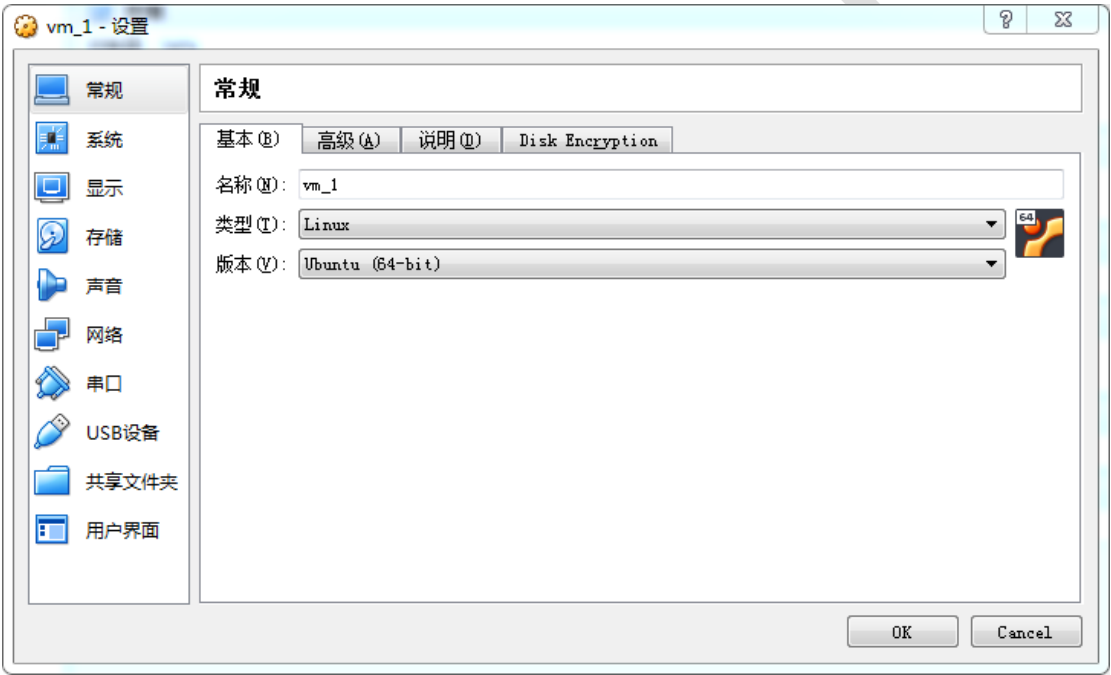


图 A-5 设置内容

设置存储，添加虚拟硬盘，依次按照红色方框进行选择。

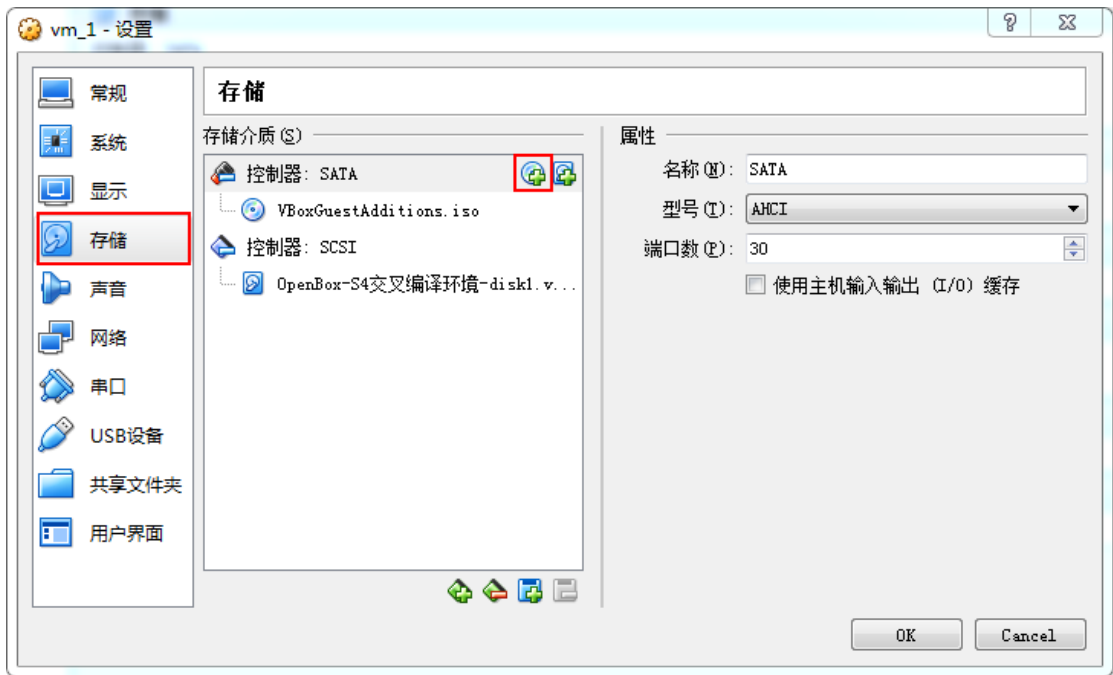


图 A-6 存储设置

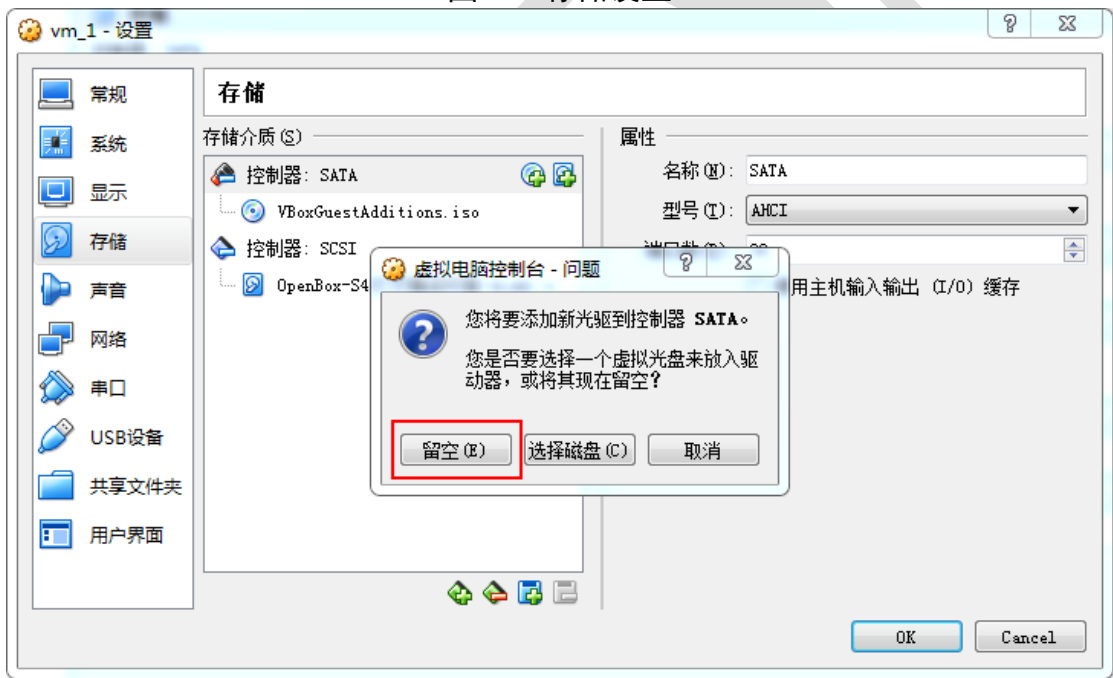


图 A-7 选择留空

设置网络，需要设置混杂模式，以便能够捕获到所有报文。

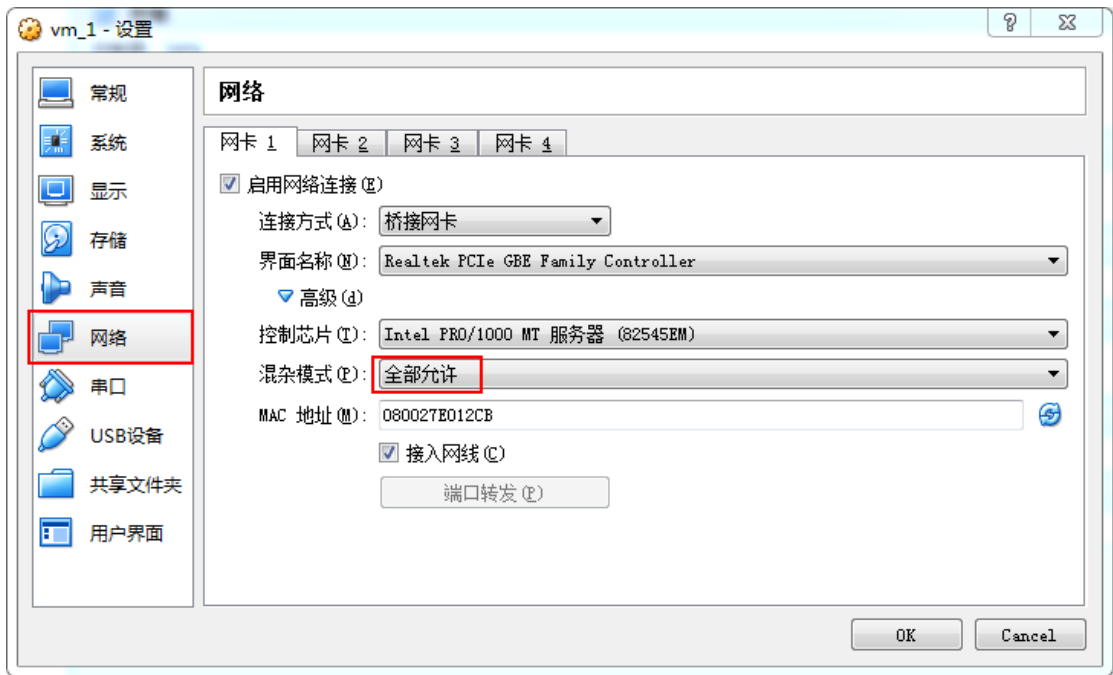


图 A-8 网卡设置

设置共享文件夹，用于 Windows 主机与虚拟机共享文件。

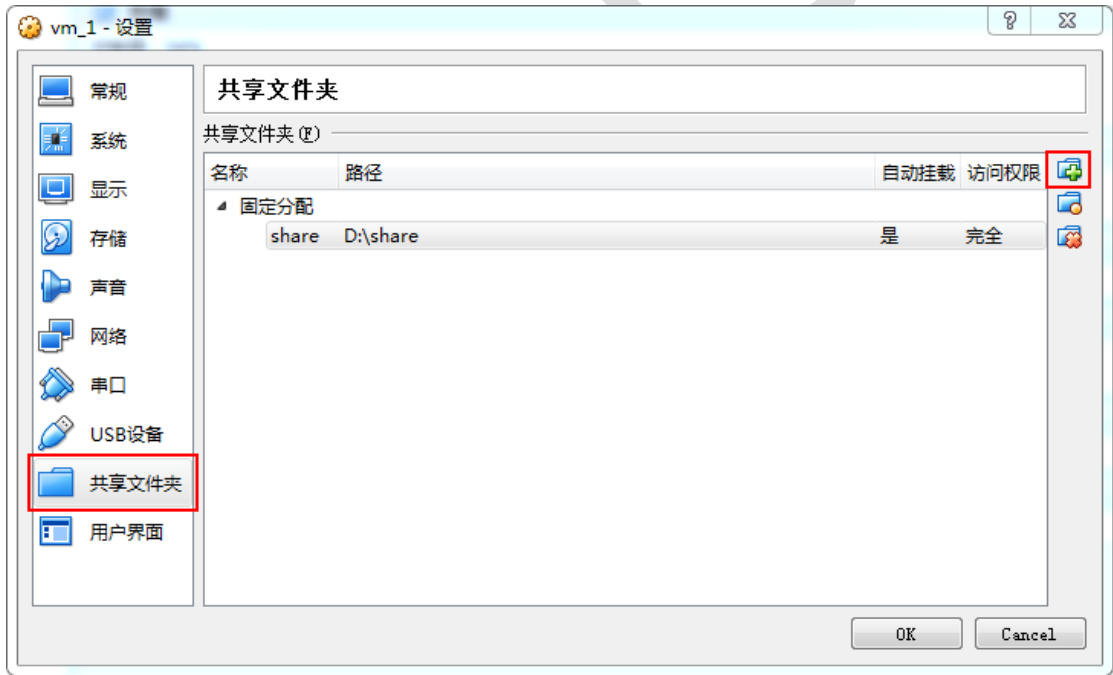


图 A-9 共享文件夹设置

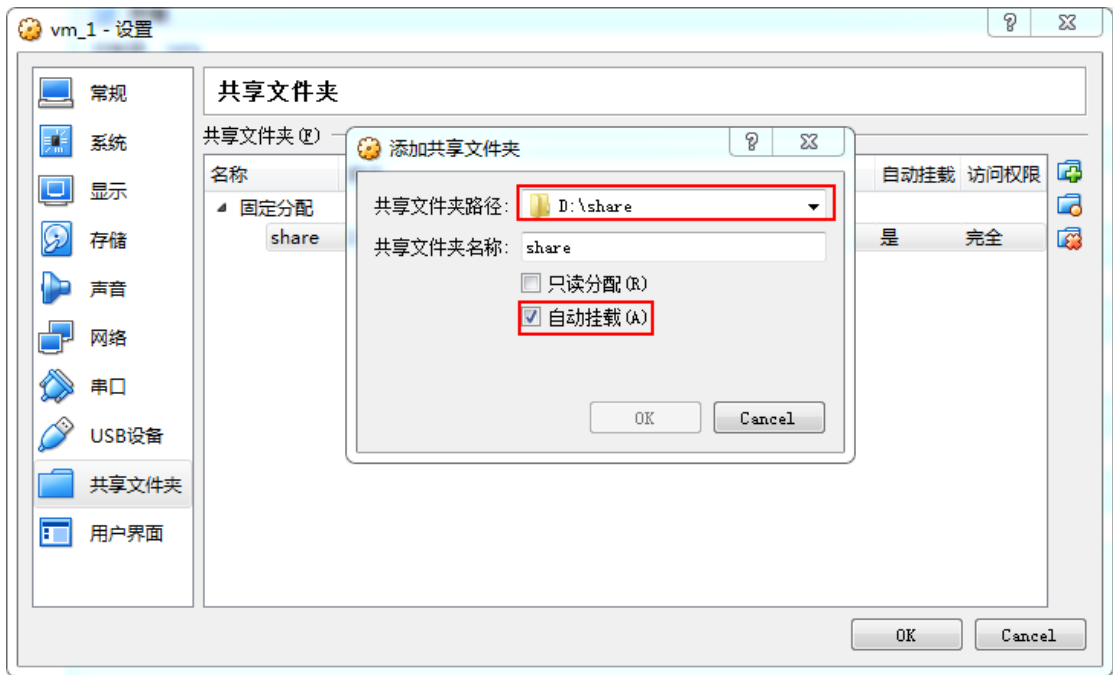


图 A-10 共享文件夹选项

(5) 进入虚拟机

然后打开虚拟机，打开时选择“无界面启动”，如正常启动电脑一样，启动后需要输入密码进入系统，密码为“970904”



图 A-11 无界面启动

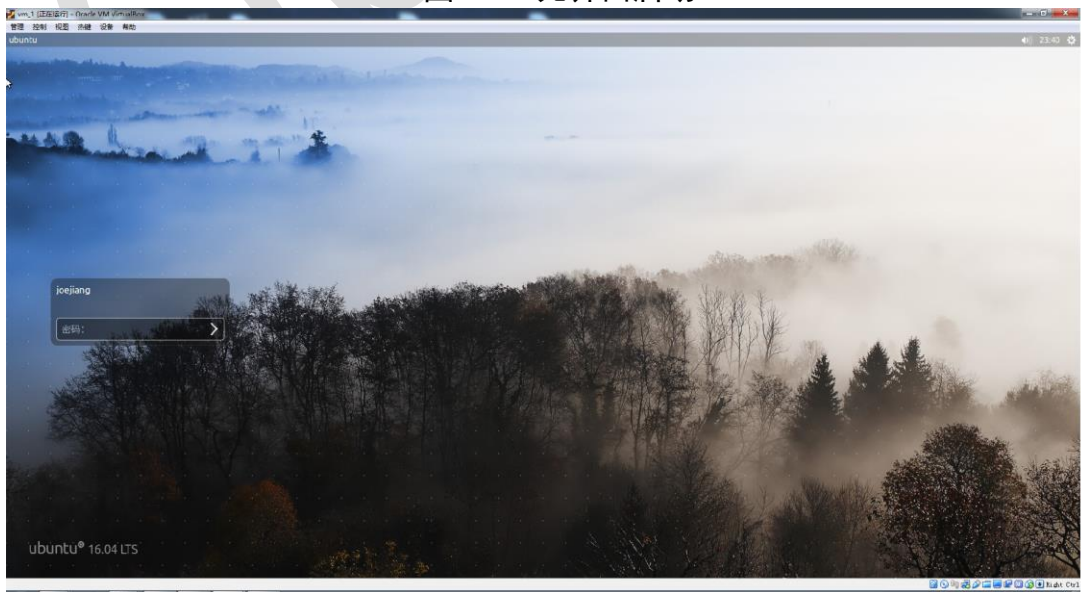


图 A-12 虚拟机启动画面

在进入系统后安装增强功能，在选择设备后，点击“安装增强功能”，然后一直“enter”键进行下一步，直至安装成功。

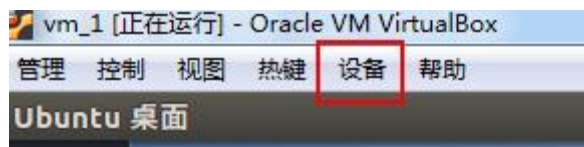


图 A-13 安装增强功能

(6) 设置共享文件夹

输入“sudo su”进入 root 权限，输入密码“970904”

输入“mount -t vboxsf share /mnt/hgfs”其中 share 为共享文件夹的名称，然后 Windows 主机与虚拟机共享 share 文件夹

输入“cd /mnt/hgfs”进入共享文件夹。

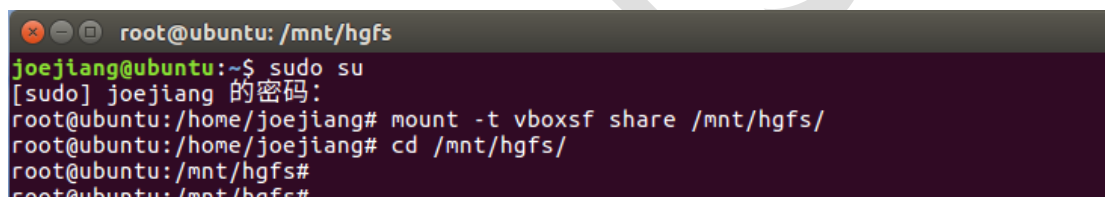


图 A-14 虚拟机共享文件夹设置

(7) 运行应用程序

把需要在虚拟机运行的应用拷贝到 Windows 主机中的 share 文件夹下，在虚拟机运行相应的应用程序。以运行 TSNLight3.2 为例介绍运行顺序。

首先输入“cd /mnt/hgfs/TSNLight3.2”进入集中控制器 TSNLight3.2 应用下；

然后输入“ifconfig”查看网卡名称，网卡名称为 enp0s17；

最后输入“./tsnlight enp0s17”执行集中控制器程序。

```

root@ubuntu: /mnt/hgfs/TSNLight3.0
root@ubuntu:~#
root@ubuntu:~#
root@ubuntu:~# cd /mnt/hgfs/TSNLight3.0
root@ubuntu: /mnt/hgfs/TSNLight3.0# ifconfig
enp0s17 Link encap:以太网 硬件地址 08:00:27:e0:12:cb
inet 地址:192.168.1.30 广播:192.168.1.255 掩码:255.255.255.0
inet6 地址: fe80::bdd0:b73f:bcb4:ad1/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
接收数据包:92064 错误:0 丢弃:0 过载:0 帧数:0
发送数据包:677 错误:0 丢弃:0 过载:0 载波:0
碰撞:0 发送队列长度:1000
接收字节:7332952 (7.3 MB) 发送字节:111271 (111.2 KB)

lo Link encap:本地环回
inet 地址:127.0.0.1 掩码:255.0.0.0
inet6 地址: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 跃点数:1
接收数据包:400 错误:0 丢弃:0 过载:0 帧数:0
发送数据包:400 错误:0 丢弃:0 过载:0 载波:0
碰撞:0 发送队列长度:1
接收字节:33520 (33.5 KB) 发送字节:33520 (33.5 KB)

root@ubuntu: /mnt/hgfs/TSNLight3.0# ./tsnlight enp0s17

```

图 A-15 集中控制器 TSNLight 运行示例

方式二：在本地的虚拟机或者 Linux 设备中运行集中控制器。

该方式不需要重新安装虚拟机，但需要在已有的虚拟机或者 Linux 设备安装软件运行时需要的库，分别为 libpcap、libnet 以及 libxml2。安装后需要设置网卡为混杂模式，以便 libpcap 能够捕获到所有报文。

注意事项：

- (1) 在运行软件时需要使用 root 权限；
- (2) 如果没有可执行文件没有权限，以 tsnlight 为例，需要赋予 tsnlight 可执行权限，输入“chmod 777 tsnlight”。

附录 2: TTE 组网拓扑

针对 opentsn3.4 的组网，硬件组网拓扑对于 TSN 或者 TTE 都是保持不变的，只需要运行不同的同步控制程序就可以在硬件架构上实

现两种时间同步方式。

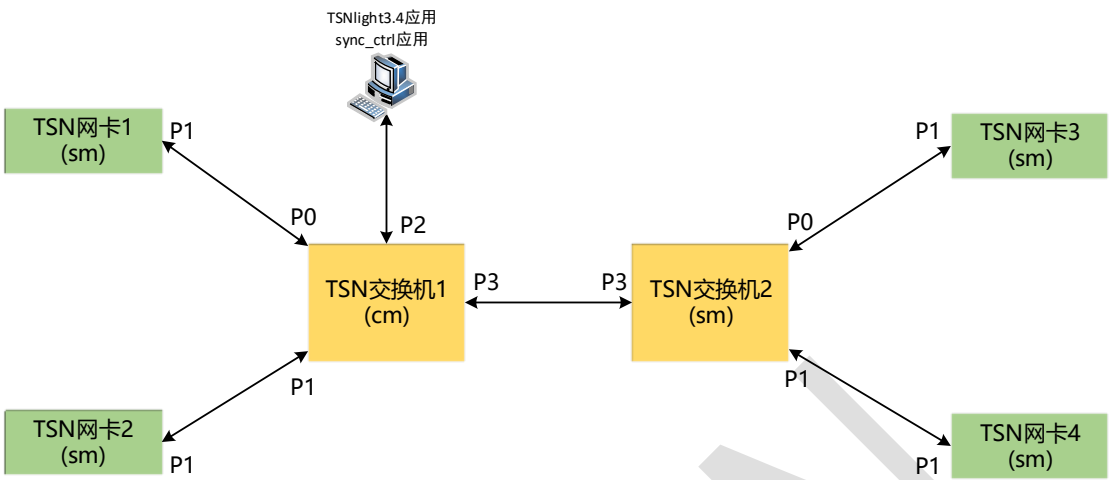


图 B-1 TTE 组网环境拓扑图