

## Application Note

## 基于 Linux® 的基本以太网接口调试



Schuyler Patton, Daolin Qiu, Alvaro Reyes

## 摘要

本应用手册说明如何解决基本板级以太网连接问题以及 TI 处理器上遇到的数据包基本发送和接收问题，其中介绍的示例参考了定制电路板或德州仪器 (TI) EVM 上的 TI Processor SDK Linux®。本应用手册重点介绍如何在 MAC 层验证无差错数据包的基本发送和接收。

## 内容

1 术语.....	2
2 先决条件.....	2
3 快速初始步骤.....	3
4 调试概述.....	4
5 网络驱动程序初始化过程.....	5
6 以太网 PHY 分析.....	8
7 以太网 MAC 统计信息分析.....	10
8 如何获取 IPv4 地址.....	12
9 跟踪数据包.....	13
10 调试网络拓扑和技术.....	14
10.1 直接连接.....	14
10.2 静态 IP 地址.....	15
10.3 设置低比特率.....	15
10.4 连接交换机时的注意事项.....	16
11 Linux® 实用程序汇总.....	17
12 申请以太网支持的检查清单.....	18

## 插图清单

图 3-1. 完整的以太网数据包流程.....	3
图 4-1. 跟踪数据包流动.....	5
图 4-2. 以太网 MAC 和以太网 PHY 接口.....	5
图 6-1. RGMII 接收多路复用和时序图.....	9
图 7-1. 具有 MAC 统计信息的完整以太网数据包流程.....	10
图 8-1. Wireshark DHCP 消息.....	12
图 9-1. DUT 发送路径.....	13
图 9-2. DUT 接收路径.....	14
图 10-1. 直接连接拓扑.....	15
图 10-2. Wireshark Ping 消息.....	16

## 商标

E2E™ is a trademark of Texas Instruments.

Linux® is a registered trademark of Linus Torvalds.

Ubuntu® is a registered trademark of Canonical Ltd.

Arm® is a registered trademark of Arm Limited.

所有商标均为其各自所有者的财产。

## 1 术语

<b>CPSW</b>	通用平台交换机 (Common Platform Switch), 这是 TI 设计的一种以太网 MAC, 通常用于通用以太网应用
<b>PRU_ICSSG</b>	可编程实时单元 - 工业通信子系统 - 千兆位 (Programmable Real-Time Unit - Industrial Communications Subsystem Gigabit), 这是 TI 设计的一种以太网 MAC, 通常用于具有严格低延迟要求的工业以太网应用
<b>PHY</b>	以太网 PHY 或以太网物理收发器
<b>MAC</b>	以太网 MAC 或以太网媒体访问控制器
<b>DTS</b>	器件树源 (Device Tree Source)
<b>CRC</b>	循环冗余校验 (Cyclic Redundancy Check), 这是以太网通信使用的一种校验和算法, 当器件接收到损坏的数据时会显示 CRC 错误
<b>移植</b>	在本应用手册的语境中, 移植是指配置 Linux 电路板 DTS 的过程
<b>DUT</b>	被测器件
<b>链路伙伴</b>	描述了直接连接到 DUT 的器件
<b>直接连接</b>	描述了具有两个直接连接平台的网络拓扑
<b>跟踪数据包</b>	本应用手册中介绍的一种调试技术, 工程师使用 MAC 硬件统计信息来跟踪以太网数据包在线路上发送和接收的过程中所经过的路径
<b>MPU</b>	微处理器单元 (Microprocessor Unit)

## 2 先决条件

下面详细列出了调试 TI EVM 或定制电路板上的以太网接口之前所需的设置：

- 当两个具有以太网接口的平台相互连接时, 被测平台或器件被称为 DUT, 而另一个平台被视为链路伙伴
- DUT 可以是定制板或 TI EVM
- 如果问题出在定制电路板上, 请在运行最新 TI SDK 的 TI EVM 上尝试同样的测试
- 为了获得最佳效果, 请为 DUT 使用最新的 TI 软件开发套件 (SDK)
- 在 TI EVM 或定制电路板上使用 Linux 用户控制台
- 执行调试的工程师知道如何使用 Linux 控制台
- 一次仅使用一个接口以保持调试尽可能简单
- DUT 连接到已知的有效链路伙伴, 如 PC 或其他经证明可正常工作的电路板
- 已配置 DTS 文件, 使 DUT 已成功引导至 Linux 用户命令提示符处
- 如果使用 Ubuntu® PC 作为链路伙伴, 请勿以虚拟机的形式运行该 PC, 因为这样生成的以太网 MAC 统计信息可能不准确

### 3 快速初始步骤

在调试潜在的网络连接问题之前，需要执行一些初始步骤来验证以太网数据包的基本发送或接收是否存在问题。第一步是查看是否定义了一个以太网接口。下一步是确定是否在两个链路伙伴之间建立了链路。最后一步是验证以太网 MAC 统计信息是否显示正在发送或接收数据包而没有发生任何损坏。

下文总结了调试以太网接口的初始步骤。请在 DUT 已成功引导至 Linux 环境后执行这些步骤。

1. 检查是否已使用 `ifconfig -a` 或 `ip link show` 初始化接口
2. 使用 `ethtool <interface name>` 确定是否与链路伙伴建立了链路
  - 例如，在 TI EVM 上，使用 `ethtool eth0` 查看第一个以太网接口的详细信息
3. 使用 `ethtool -s <interface name>` 验证是否发送或接收了任何数据包

链路伙伴一词就是指两个直接连接的平台，如图 3-1 所示。这称为直接连接拓扑。对于上述三个初始步骤，为 DUT 推荐的链路伙伴是运行 Ubuntu 的 PC 或另一个已知运行良好的 Linux 平台。推荐链路伙伴采用 Linux 平台的原因是为了提供一个对称测试环境，以便链路伙伴与运行 Linux 的 DUT 具有类似平台。此外，在 Linux 平台上可以轻松使用 `ethtool`。

图 3-1 显示了以太网接口如何细分为两个部分：MAC 和 PHY。MAC 负责发送和接收数据包。PHY 负责将信号从 MAC 转换到以太网电缆。

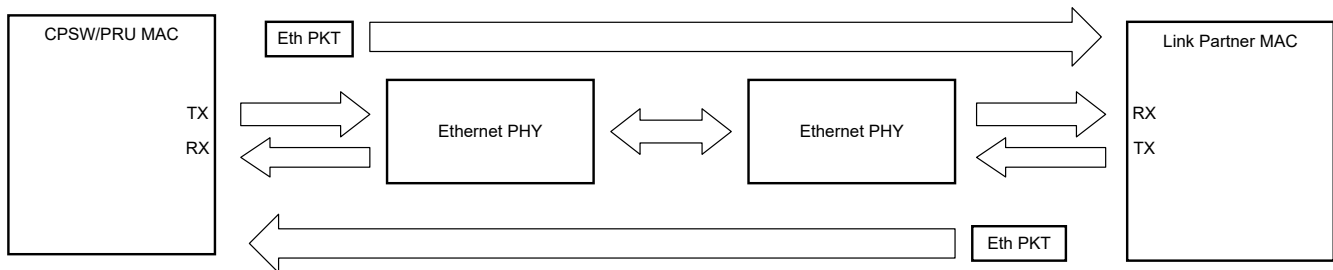


图 3-1. 完整的以太网数据包流程

确认使用 `ifconfig -a` 或 `ip link show` 列出了要测试的接口后，下一步是查看是否检测到与链路伙伴的以太网链路。

#### 备注

调试建议：

如果 `ifconfig` 或 `ip a` 命令未列出该接口，这可能意味着定义 CPSW 或 PRU-ICSSG 以太网接口的 DTS 文件中存在错误。CPSW 和 PRU-ICSSG 都是 TI 设计的以太网接口。另请参阅节 5。

当两个链路伙伴（例如 TI EVM 和 PC）之间连接了以太网电缆时，这两个器件上的以太网 PHY 会尝试在这两个器件之间建立链路，此过程不依赖于运行的任何软件（例如 ping 测试）。如果检测到链路，下一步是检查是否有任何接收数据包或发送数据包的流量。如果 DUT 上的以太网驱动程序没有检测到链路，则以太网驱动程序不会初始化，也不会传递网络流量。

`ethtool` 实用程序用于执行链路检测和流量分析这两个步骤。步骤 1 是检查链路检测。在此示例中，被测接口假定为 `eth0`。以下代码片段显示了 `ethtool` 命令的部分输出。打印出来的信息有很多，但需要关注的部分是 `Link detected:` 属性。`Linked detected:` 为 `yes`（如果检测到链路）或 `no`（如果未检测到链路）。此命令还返回链路速度和双工模式信息。

```
root@am62xx-evm:~# ethtool eth0 Settings for eth0: ... Speed: 1000Mb/s Duplex: Full ... PHYAD: 0
Link detected: yes
```

如果未检测到链路，则表示无法传递流量。请先解决此问题，然后再继续。

## 备注

调试建议：

如果已连接的 PHY 未检测到链路，则下一步是查看[节 6](#)中说明的 PHY 硬件配置和 MDIO 通信信息。

检测到以太网链路后，可以查看 MAC 硬件统计信息 (`ethtool -s <interface name>` 的结果)。从 MAC 硬件统计信息返回的信息包含有关基本数据包发送和接收的数据。通过检查 DUT 上是否发送和接收了数据包，即可了解以太网接口和以太网 PHY 是否正常工作。此外，DUT 未收到 IPv4 地址或未收到 ping 响应并不意味着数据包的基本发送和接收存在问题。例如，未收到 IPv4 地址的原因可能很简单，只是因为未设置 DHCP 服务器来提供 IPv4 地址。

就以太网 MAC 是否正在发送和接收数据包而言，以太网 MAC 统计信息是最准确的一组统计信息。之所以说这些统计信息是可查看的最准确的统计信息，是因为此数据直接从物理线路测得，没有额外的抽象层。从 MAC 的角度来看，`ifconfig` 或 `ip -s link` 命令显示的已发送数据包 (TX 数据包) 或已接收数据包 (RX 数据包) 计数并不完全准确。在内核网络协议栈和 MAC 驱动程序之间可能会丢弃数据包。

以下代码片段显示了以太网 MAC 硬件统计信息的部分示例，其中显示了接收和发送的数据包以及所有相关的错误。此示例显示了在检测到正确接收以太网帧和成功发送以太网帧时的预期结果。可能存在某些仅可通过连接的链路伙伴以太网 MAC 硬件统计信息来查看的发送错误。有关链路伙伴分析的更多详细信息，请参阅[节 9](#)。

```
root@am62xx-evm:~# ethtool -s eth0 NIC statistics: ... rx_good_frames: 1127 ... rx_crc_errors: 0
rx_align_code_errors: 0 ... tx_good_frames: 163
```

有关如何分析 RX 和 TX 流量错误的更多信息，请参阅[节 7](#)。

总之，如果定义了网络接口 (例如 `eth0`)，检测到与链路伙伴之间存在链路，发送和接收数据包计数器在递增，并且 MAC 统计信息中没有错误计数器在递增，那么以太网数据包的基本发送和接收没有问题。

一个常见问题为是否需要诸如 ping 或 `iperf3` (以前名为 `iperf`) 之类的应用程序来执行以太网帧的基本发送和接收。Linux 引导完成后，会尝试使用广播消息获取 IP 地址。来自这些广播消息的流量用于执行基本的以太网帧发送和接收。有关如何获取 IPv4 地址的更多详细信息，请参阅[节 8](#)。

## 备注

调试建议：

- 如果没有发送流量，可能有多种原因导致出现这种情况，例如无 PHY 时钟、引脚多路复用 (PinMuxing) 等。另请参阅[节 6](#)。
- 如果没有接收流量，可能是多种原因之一导致的，例如时钟、数据时序、引脚多路复用 (PinMuxing) 等。另请参阅[节 6](#)，并查看相关 DTS 属性。
- 查看[以太网 MAC 统计信息分析](#)。

章节摘要：

- 使用 `ethtool <interface name>` 检查所需接口是否有链路。如果没有链路，则必须调试该问题。如果没有检测到链路，任何数据都无法传递。
- 如果检测到链路，则可以使用 `ethtool -s <interface name>` 来检测以太网帧 (或数据包) 基本发送和接收 (即查看以太网 MAC 硬件统计信息)。如果检测到以太网帧或数据包，则表明接口正在基本层面工作。如果统计信息表明没有发送或接收过以太网帧或数据包，则可以对这方面进行调试。

## 4 调试概述

[节 3](#) 讨论了一些初始调试检查，旨在确定数据包的基本发送和接收状态。这里介绍了一种称为跟踪数据包的技术，该技术使用以太网 MAC 接口统计信息来跟踪以太网数据包在线路上发送和接收的过程中所经过的路径。[图 4-1](#) 说明了发送和接收数据包的基本要求。本应用手册的其余部分将演示如何分析方框图的每个阶段。

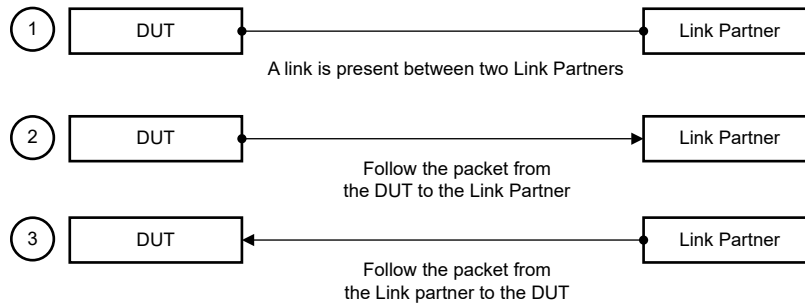


图 4-1. 跟踪数据包流动

两个链路伙伴之间数据包流动的基本步骤：

1. 只有在两个链路伙伴之间建立以太网链路后，数据包才能流动。在建立链路的过程中，通常采用自动协商来确定最佳的位线速率和双工模式。请注意，自动协商在以太网 PHY 之间进行物理连接时会触发。除了建立链路之外，必须已针对以太网 MAC、MDIO 接口和相应的以太网 PHY 正确初始化 Linux 驱动程序。另请参阅 [网络驱动程序初始化过程](#)。
2. 建立链路后，可以独立检查发送路径。本应用手册介绍了如何使用 MAC 硬件统计信息来确保数据包已通过线路发送并在 MAC 层由链路伙伴接收到。在此层级使用何种协议无关紧要，只发送以太网数据包。
3. 与发送路径一样，可以独立检查接收路径。本应用手册通过查看从链路伙伴接收数据包的 DUT 以太网接口上的 MAC 硬件统计信息，展示了如何在 MAC 层接收数据包。

图 4-2 显示了以太网 MAC 和以太网 PHY 之间的连接。这里假设读者了解以太网 MAC 和以太网 PHY 的功能和用途。以太网 MAC 负责数字数据的发送和接收，PHY 负责数字域和模拟域之间的以太网数据包转换。

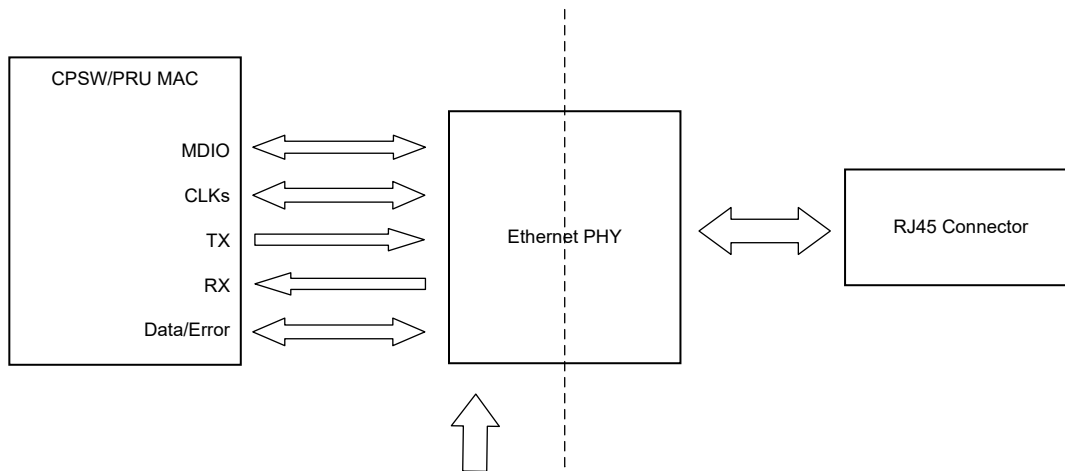


图 4-2. 以太网 MAC 和以太网 PHY 接口

图 4-2 中的元件显示了以太网 MAC 和以太网 PHY 之间的数据路径 (RX、TX)、控制路径 (MDIO) 以及错误路径或同步路径。这些是 MAC 和 PHY 接口之间所需的常用信号。根据 MAC 和 PHY 之间的接口模式 (MII、RGMII、RGMII 等)，这些信号存在不同的变体。

## 5 网络驱动程序初始化过程

在 TI 基于 Arm® 的处理器上，有两个在 Linux 引导期间初始化的驱动程序：MDIO 驱动程序以及 CPSW 和/或 PRU\_ICSSG 驱动程序。MDIO 驱动程序用于与以太网 PHY 通信。CPSW 和/或 PRU\_ICSSG 驱动程序可以设置以太网 MAC。为了检查这些驱动程序是否已正确初始化，需要检查 DUT 的引导日志。此时将出现以下初始化阶段：

1. MDIO 和 CPSW 或 PRU\_ICSSG 驱动程序均已初始化且没有错误。
2. MDIO 驱动程序已在电路板 DTS 文件中定义的相应 PHY 地址处检测到 PHY。确切的 PHY 地址由 PHY 硬件自举决定。

3. 以太网接口为 up 状态并建立了链路，假设在启动前已将电缆连接到以太网端口。

以下日志显示了初始化过程中应该执行的操作的示例。为每行打印的时间戳在不同的引导序列之间可能不同。

```
[ 1.060850] davinci_mdio 8000f00.mdio: Configuring MDIO in manual mode
[ 1.095345] davinci_mdio 8000f00.mdio: davinci mdio revision 9.7, bus freq 1000000
[ 1.098179] davinci_mdio 8000f00.mdio: phy[0]: device 8000f00.mdio:00, driver TI DP83867
[ 1.098198] davinci_mdio 8000f00.mdio: phy[1]: device 8000f00.mdio:01, driver TI DP83867
[ 1.098246] am65-cpsw-nuss 8000000.ethernet: initializing am65 cpsw nuss version 0x6BA01103,
cpsw version 0x6BA81103 Ports: 3 quirks:00000006
[ 1.098500] am65-cpsw-nuss 8000000.ethernet: initialized cpsw ale version 1.5
[ 1.098505] am65-cpsw-nuss 8000000.ethernet: ALE Table size 512
[ 1.099141] pps pps0: new PPS source ptp0
[ 1.099456] am65-cpsw-nuss 8000000.ethernet: CPTS ver 0x4e8a010c, freq:500000000, add_val:1 pps:1
[ 1.120288] am65-cpsw-nuss 8000000.ethernet: set new flow-id-base 19
```

在此示例中，MDIO 驱动程序是 `davinci_mdio`，CPSW 驱动程序是 `am65-cpsw-nuss`。DUT 上有两个 TI DP83867 PHY，预期的 PHY 地址将自举至地址 0 和 1。MDIO 驱动程序在电路板 DTS 中指定的这些地址处检测到这些 PHY，并将 PHY 连接到 DP83867 PHY 驱动程序。在启动序列之前未连接以太网电缆，因此不会显示链路 up 状态消息。

以下日志显示了 MDIO 驱动程序扫描总线以查找在 DUT 的 DTS 文件中定义的以太网 PHY。这是符合预期的正常初始化序列。此过程将确定 PHY 供应商，也将确定相应的 PHY 驱动程序。

```
[ 6.902614] am65-cpsw-nuss 8000000.ethernet eth1: PHY [8000f00.mdio:01] driver [TI DP83867]
(irq=POLL)
[ 6.902648] am65-cpsw-nuss 8000000.ethernet eth1: configuring for phy/rgmii-rxid link mode
[ 6.938874] am65-cpsw-nuss 8000000.ethernet eth0: PHY [8000f00.mdio:00] driver [TI DP83867]
(irq=POLL)
[ 6.938911] am65-cpsw-nuss 8000000.ethernet eth0: configuring for phy/rgmii-rxid link mode
```

电路板 DTS 文件中定义的 PHY 然后由相应 DTS 定义中的 CPSW 端口引用。以下代码是从 TI EVM 提取的一段编辑过的代码，其中展示了 MDIO 和用于以太网端口的 PHY，以及在 Linux 引导期间如何在 Linux 控制台中识别 MDIO 和 PHY。

```
A DTS PHY node example, shortened for discussion purposes
&cpsw3g_mdio {
    cpsw3g_phy0: ethernet-phy@0 {
        reg = <0>;    <-- check PHY address, verify this is correct
        .....
    };
};

This line shows the PHY defined in the DTS being identified
[ 6.938874] am65-cpsw-nuss 8000000.ethernet eth0: PHY [8000f00.mdio:00] driver [TI DP83867]
(irq=POLL)
```

如果 MDIO 驱动程序无法与以太网 PHY 通信，则可能会发生多个错误，这些错误要么与 DTS 有关，要么由于硬件实施问题导致 PHY 状态出现问题。

### 备注

针对 PHY 初始化问题的调试建议：

- 确认 DTS 中的 PHY 节点正确
- 另请参阅 [以太网 PHY 分析](#)

现在，以太网 PHY 已经被识别，接下来可以查询这些 PHY 以查看是否与链路伙伴建立了链路。如果连接了以太网电缆并建立了链路，DUT 控制台上会显示以下日志。数据比特率和双工模式在建立链路时商定。请注意，在此示例中，速度为 1Gbps、链路为全双工、以太网接口随时可用。

```
[ 11.042192] am65-cpsw-nuss 8000000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 11.042255] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

TI EVM 上的以太网速度和双工模式在链路伙伴之间通过自动协商确定。上面的消息显示了一个成功的初始化序列。

如果驱动程序初始化有问题，建议参考与电路板设计相关的 TI EVM，检查电路板 DTS。以下代码是用于定义 TI SK-AM62B-P1 EVM 的 CPSW 接口 (以太网 MAC) 的 DTS 文件示例，可用作参考。

以下 DTS 示例展示了在使用定制电路板 DTS 时需要验证的一些检查点。

```
Linux Source tree directory: arch/arm64/boot/dts/ti/k3-am62x-sk-common.dtsi
...
&cpsw3g {
    bootph-all;
    pinctrl-names = "default";
    pinctrl-0 = <&main_rgmii1_pins_default>;    <-- verify pin mux is correct
};

&cpsw_port1 {
    bootph-all;
    phy-mode = "rgmii-rxid";    <-- verify interface mode is correct
    phy-handle = <&cpsw3g_phy0>;    <-- verify phy name is referenced correctly
};
...
&cpsw3g_mdio {
    bootph-all;
    status = "okay";
    pinctrl-names = "default";
    pinctrl-0 = <&main_mdio1_pins_default>;    <-- verify pin mux is correct

    cpsw3g_phy0: ethernet-phy@0 {
        bootph-all;
        reg = <0>;
        ti,rx-internal-delay = <DP83867_RGMIIDCTL_2_00_NS>; <-- rx clock delay
        ti,fifo-depth = <DP83867_PHYCR_FIFO_DEPTH_4_B_NIB>;
        ti,min-output-impedance;
    };
};
```

章节摘要：

以下是表明初始化过程成功的条件：

- MDIO 驱动程序已初始化且无错误，并且已识别 MDIO 总线上的所有预期 PHY。
- CPSW 和/或 PRU\_ICSSG 驱动程序已初始化且无错误。
- CPSW 驱动程序报告了一条包含预期比特率和双工模式的链路 up 状态消息。

## 6 以太网 PHY 分析

本节介绍一些基本的以太网 PHY 调试步骤，旨在确保 PHY 已初始化并正确运行。

要检查的第一个区域是以太网 PHY 的电源和时钟输出信号。如果可能，请查阅以太网 PHY 供应商的故障排除指南。例如，TI DP83869 以太网 PHY 的故障排除指南演示了如何对 DP83869 PHY 进行基本评估。

出于讨论目的，这里以 [DP83869 故障排除指南](#) 作为以太网 PHY 电压和时钟基本调试的示例。

以太网 PHY 上有一些外设引脚需要密切关注。参考故障排除指南的 [外设引脚检查](#) 部分有这些引脚的具体说明。参考故障排除指南中还提供了更多详细信息。

- 电源
  - 确认每个电压轨位于制造商以太网 PHY 数据表指定的容差范围内
- 探测 XI 时钟
  - 输入时钟是 PHY 正常运行的必要条件
  - 确认信号存在且满足 PHY 的规格
- 探测 RESET\_N 信号
  - 确认 PHY 未保持在复位状态
- 在初始化期间探测配置 (Strap) 引脚
  - PHY 通过这种方式查找配置
    - 示例：配置之一是 MDIO 总线上的 PHY 地址
  - 故障排除指南推荐了应遵循的特定调试技术
  - 另请参阅 TI E2E™ 常见问题解答：[如何确认以太网 PHY 配置 \(Strap\)](#)
- 探测串行管理接口信号 (MDC、MDIO)
  - 参考 PHY 数据表中的预期电阻容差和信号完整性要求



- 如果可以通过 `ethtool` 实用程序 (例如 `ethtool eth0`) 访问 PHY 寄存器, 则无需执行此步骤

如果已验证外设引脚正确并且 MDIO 总线正确发送信号, 但是未与 PHY 进行通信, 那么下一个调试步骤是检查 MDIO 驱动程序和所选的 PHY 驱动程序。节 5 介绍了对 MDIO 驱动程序和通信的分析。在 [如何集成 Linux 驱动程序](#) 应用手册的 `dmesg | grep -i mdio` 部分提供了一个额外的参考, 其中介绍了一个 MDIO 交互示例。

处于 RGMII 模式时, 请特别注意 MAC - PHY 接口。这里的关键调试元素是从 PHY 到 MAC 的 RX 时钟需要相对于 RX 数据延迟。如果该时序不正确, 则会导致以太网帧的接收出现问题。诸如丢帧或 RX CRC 错误等错误将在 MAC 层被测量到。在 RGMII 接口模式下运行时, TI 基于 Arm® 的处理器 MAC 会自动对 TX 时钟施加相对于 TX 数据的 2ns 延迟。另一个注意事项是电路板布局布线以及时钟和数据线布线长度对于定义正确的时钟延迟至关重要。 [高速接口布局指南](#) 应用手册提供了有关高速布局重要性的信息。

电路板 DTS 定义了由 PHY 驱动程序使用的 RX 时钟延迟信息。有关如何配置 RX 时钟延迟的示例, 请参阅 [如何集成 Linux 驱动程序](#) 应用手册。

以下代码片段是关于如何配置 RX 时钟延迟的 DTS 示例。

```
(Linux source tree directory) arch/arm64/boot/dts/ti/k3-am62x-sk-common.dtsi (a portion from this file)
#include <dt-bindings/net/ti-dp83867.h> <-- Has the macro defines used in the phy node

cpsw3g_phy0: ethernet-phy@0 {
bootph-all;
reg = <0>;
ti,rx-internal-delay = <DP83867_RGMIIDCTL_2_00_NS>; <-- rx clock delay
ti,fifo-depth = <DP83867_PHYCR_FIFO_DEPTH_4_B_NIB>;
ti,min-output-impedance;
};
```

此处所需的 RX 时钟延迟的示例参考了 DP83869 以太网 PHY 数据表, 其中定义了必要的 RX 时钟延迟。

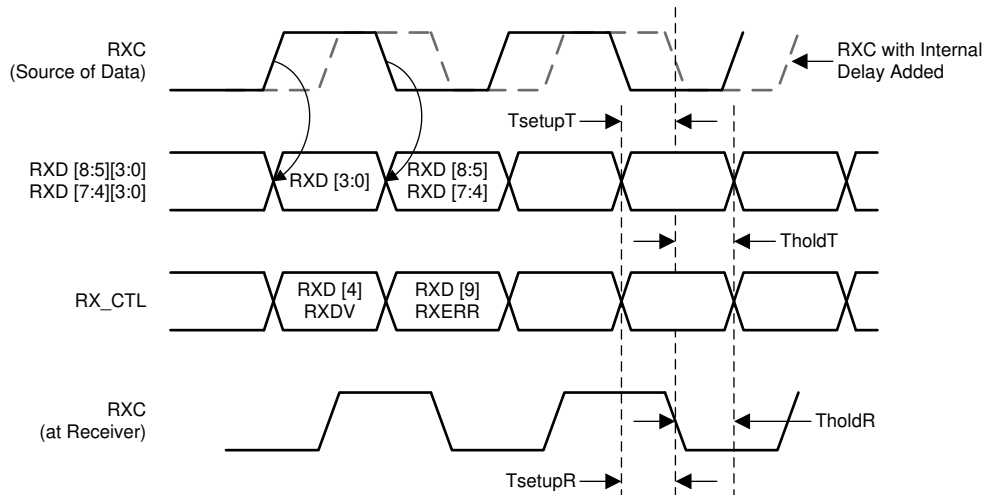


图 6-1. RGMII 接收多路复用和时序图

章节摘要：

如何验证以太网 PHY 是否在运行

- 验证外设引脚是否按预期工作，以及是否在 PHY 数据表和 PHY 故障排除指南（如果存在）规定的容差范围内。
- 确认 MDIO 驱动程序已初始化且没有错误，并且已在 Linux 控制台日志中识别出所有预期的 PHY。
- 如果 PHY 接口为 RGMII，请检查 RX 时钟是否根据电路板布局布线延迟，并验证在电路板 DTS 文件中是否正确配置了 RX 时钟延迟。

## 7 以太网 MAC 统计信息分析

通过查看 MAC 统计信息可以了解 DUT 和链路伙伴之间是否存在错误，从而验证以太网数据包的基本发送和接收是否成功。请注意，即使发生了数据包的基本发送和接收，也可能存在限制网络吞吐量的错误。本节将讨论一些错误和解决方案。

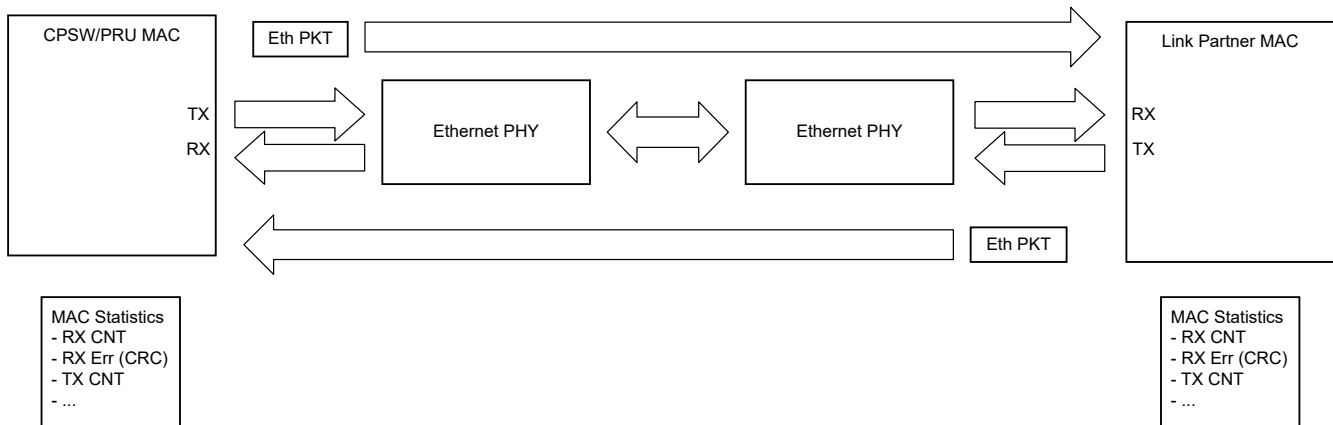


图 7-1. 具有 MAC 统计信息的完整以太网数据包流程

通过查看 MAC 统计信息可以了解发送、接收的数据包数以及因错误而丢弃的数据包数。请注意，MAC 统计信息是直接接口上监视数据包的结果。

检查接口统计信息时无需分析单个数据包的粒度。需要注意的主要行为是 RX 和 TX 正常帧计数递增，以及确认未检测到 CRC、对齐等错误。

递增 TX 和 RX 正常帧统计信息寄存器表明在 MAC 和 PHY 之间发送和接收数据包的行为正常。这也表明 MAC-PHY TX 和 RX 时钟的行为正常。

如果 MAC 统计信息错误计数器不断增加，则表明数据包在接口上不断丢弃。例如，RX CRC 错误计数器为非零值表明接收到的数据包具有 CRC 校验和错误，因此在接口上丢弃了这些数据包。

### 备注

CRC 错误等 TX 错误不在可捕获的被测平台 MAC 统计信息列表中。TX CRC 错误仅由链路伙伴的 RX 错误计数器进行检测。

以下 MAC 统计信息适用于 TI EVM 直接连接另一电路板（通过在两个链路伙伴之间连接以太网电缆）的情况。两个链路伙伴的任一链路都没有运行 DHCP 服务器来动态分配 IP 地址。如节 3 所述，无需获取 IP 地址即可检查数据包流量。在检查此层级的流量时，关键在于确定在基本数据包发送和接收期间没有观察到错误。

一旦检测到接口上有链路为 up 状态并初始化了 MAC 驱动程序，便会启动网络通信以尝试获取 IP 地址。以下 TX 和 RX 就是此网络通信的结果，表示已发送和接收的数据包数量。

具体来说，MAC 统计信息中的 TX 数据包数量的示例信息片段是两个链路伙伴之间连接以太网电缆的结果。一旦检测到接口上有链路为 up 状态，Linux 内核便会尝试获取 IP 地址。tx\_broadcast\_frames 统计信息中报告的一些

帧是来自 DUT 的 DHCP 请求帧，用于获取 IPv4 地址。此外，如果 `tx_good_frames` 统计信息在递增，这充分表明数据包已从 DUT 的 MAC 发送到线路上。

A portion of an example `ethtool -S <interface>` output showing the TX MAC statistics:

```
tx_good_frames: 19
tx_broadcast_frames: 2
tx_multicast_frames: 17
```

根据 MAC 统计信息中的以下 RX 结果可以看出，在连接以太网电缆后也接收到数据包。这些特定统计信息表明 MAC 正确接收了一些数据包。在此示例中，没有为平台分配 IP 地址。在此层级，使用的实际以太网协议并不重要。主要目标是确认未接收到损坏的数据包以及正确发送了数据包。

A portion of an example `ethtool -S <interface name>` output showing RX MAC statistics:

```
rx_good_frames: 104
rx_broadcast_frames: 10
rx_multicast_frames: 94
```

下一个示例详细介绍了 MAC 统计信息中的 RX 错误结果。这里没有检测到错误。但是，如果这些错误统计信息在递增，则表明存在导致数据包在接口上被丢弃的特定错误。对于用作 DUT 的德州仪器 (TI) 处理器，每个 MAC 统计信息计数器的定义可在特定处理器的相应技术参考手册 (TRM) 中找到。

Portion of sample `ethtool -S <interface name>` showing RX error statistics:

```
rx_crc_errors: 0
rx_align_code_errors: 0
rx_oversized_frames: 0
rx_jabber_frames: 0
rx_undersized_frames: 0
```

---

### 备注

调试建议：

- 如果 `ethtool` 实用工具未返回任何链路检测数据或 MAC 统计数据，则表示与 PHY 的 MDIO 通信未按预期执行。确认 PHY 器件驱动程序已加载并在 Linux 引导日志中正确识别。如果未正确识别 PHY，请参阅节 6。
- 如果有 TX 流量，但没有 RX 流量，请参阅节 4、节 9 或本应用手册每个章节末尾的调试建议。

章节摘要：

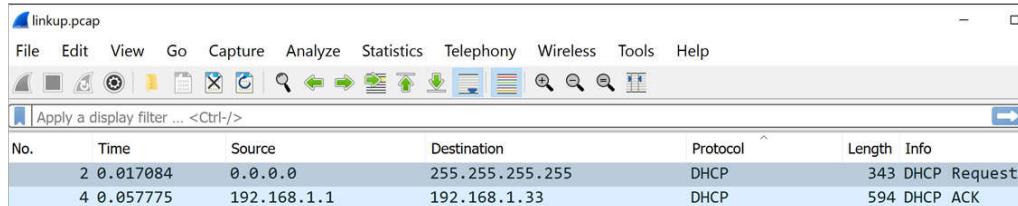
- 接口的 `ethtool -S <interface name>` 结果提供了有关发送和接收过程的数据。
- 在两个链路伙伴之间连接以太网电缆后生成的流量可通过 `ethtool` 查看，无需获取 IP 地址。

## 8 如何获取 IPv4 地址

虽然不需要成功获得分配给被测 DUT 以太网接口的 IPv4 地址，但获取 IPv4 地址的过程提供了一种测试数据包基本发送和接收的方法。在默认的 TI SDK 文件系统中，当检测到以太网链路已建立时，便会尝试获取 IP 地址。未收到 IPv4 地址时，大多数用户都认定网络连接存在问题。不过，这种情况在网络拓扑中设置了 DHCP 服务器时会出现。如果未设置 DHCP 服务器，那么未收到 IPv4 地址并不总是表明网络连接失败，只是没有设置服务器来分配 IPv4 地址。

获取 IPv4 地址的第一步是确保 DUT 连接到具有 DHCP 服务器的链路伙伴或具有 DHCP 服务器的网络。如果没有 DHCP 服务器，则此 IP 地址获取过程将失败，并显示被测平台没有发送或接收数据包。

图 8-1 显示网络驱动程序初始化后，网络管理器会发送 DHCP 请求来获取 IP 地址。DHCP 服务器提供响应或 ACK，其中包含其分配给 DUT 的 IP 地址。



No.	Time	Source	Destination	Protocol	Length	Info
2	0.017084	0.0.0.0	255.255.255.255	DHCP	343	DHCP Request
4	0.057775	192.168.1.1	192.168.1.33	DHCP	594	DHCP ACK

**PKT 2**

```

> Frame 2: 343 bytes on wire (2744 bits), 343 bytes captured (2744 bits)
> Ethernet II, Src: 08:04:b4:32:5d:ce (08:04:b4:32:5d:ce), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Dynamic Host Configuration Protocol (Request)
                    
```

- This wireshark capture shows the DHCP request and ACK or response
- PKT 2 is the request message, this a broadcast message to the DHCP server and contains the MAC address of the device requesting an IPv4 address
- PKT 4 is the ACK response from the DHCP server with the IP address to use.

**PKT 4**

```

Client IP address: 0.0.0.0
Your (client) IP address: 192.168.1.33
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: 08:04:b4:32:5d:ce (08:04:b4:32:5d:ce)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (ACK)
                    
```

图 8-1. Wireshark DHCP 消息

如果此 IP 地址获取过程未能成功完成，则下一步是跟踪数据包从发送数据包的 DUT 到接收数据包的链路伙伴的路径。这种跟踪数据包 (FTP) 的过程也涉及链路伙伴的响应发送到 DUT 以确认接收到发送的数据包。节 9 描述了此 FTP 过程的分析。

## 9 跟踪数据包

在两个链路伙伴之间建立以太网链路后，即可跟踪数据包从 DUT 发送到链路伙伴，然后从链路伙伴发送回 DUT 的步骤。仅当在链路两端都可以访问以太网 MAC 统计信息的情况下才有效。如果网络拓扑包含直接连接（如图 3-1 所示），还可以进一步简化。

如节 8 中所述，无需设置 IP 地址即可启动网络通信。如果在 DUT 上使用 TI 默认文件系统，则会在检测到链路后尝试获取 IP 地址。此网络通信足以跟踪是否成功接收到从 DUT 发送到链路伙伴的数据包。来自链路伙伴的广播数据包足以证明在 DUT 上未接收到损坏的数据包。

在 DUT 和链路伙伴之间连接以太网电缆之前，请使用 `ethtool` 验证以太网 MAC 统计信息中的 RX 和 TX 计数器是否为 0。建议在链路伙伴平台上使用 Linux® 操作系统 (OS)。这个平台可以是另一个正常工作的 EVM 或 Ubuntu PC。`ethtool` 仅适用于运行 Linux 操作系统的平台。

如图 9-1 所示，首先检查链路伙伴是否接收到从 DUT 发送的数据包，方法是在链路伙伴上使用 `ethtool` 并检查 RX 计数器。如果链路伙伴未运行于 Linux 操作系统上，则 `ethtool` 不可用。RX 计数器可以显示接收到多少个正常帧，更重要的是接收到多少个损坏的帧。如果接收到的数据包没有错误，则表示 DUT 发送路径可以正常工作。如果链路伙伴上的 RX 错误计数器为非零值（例如，RX CRC 错误），则可以得出结论：DUT 的发送路径存在问题，需要进行调查。

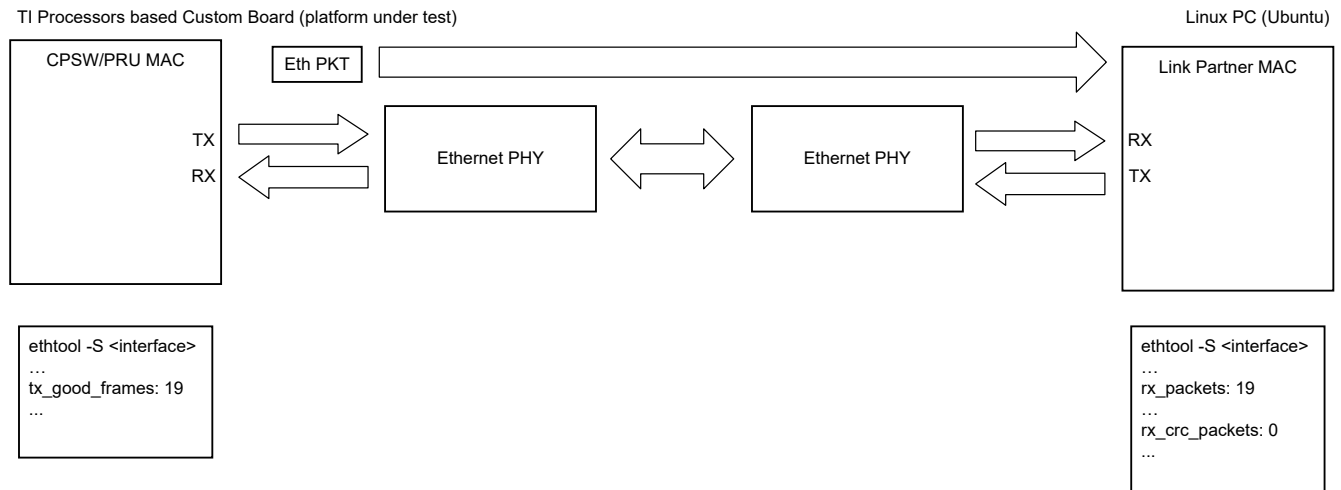


图 9-1. DUT 发送路径

作为接收方的链路伙伴通常可以观察到两个问题：未接收到数据包或 CRC 错误。如果链路伙伴未接收到任何数据包，则表明从 DUT 发出的以太网帧受到了损坏，以至于 MAC 接口无法识别该帧。当未检测到数据包或检测到 CRC 错误时，需要更仔细地检查 DUT 上的 MAC 或 PHY 连接。

### 备注

调试建议：

- 当问题在于链路伙伴未接收到数据包时
  - 在 DTS 文件中查看并验证 PinMux
  - 验证 DTS 中的 PHY 接口模式（例如 RMII 或 RGMII）与原理图是否一致
- 如果链路接收到 CRC 或其他错误：
  - 尝试更换以太网电缆以防电缆存在损坏
  - 尝试另一个已知正常的可访问以太网 MAC 统计信息的链路伙伴
  - 如果 PHY 采用 RGMII 模式，并且 PHY 未增加 RX 时钟延迟，请增加此延迟。TI 基于 Arm® 的处理器已设置 2ns 的 TX 时钟延迟，因此 PHY 不会向 TX 时钟增加任何延迟。
  - 有关如何分析发送错误的详细信息，请查阅节 6

分析 DUT 接收路径与分析 DUT 发送路径的方式类似。观察 图 9-2 中的接收数据包计数和可能发生的一些错误。如前文所述，当错误计数器为零并且接收到的正常帧计数增加时，表示接收路径工作正常。在此层级，数据包的类型无关紧要，只接收数据包而不检测错误数据包。

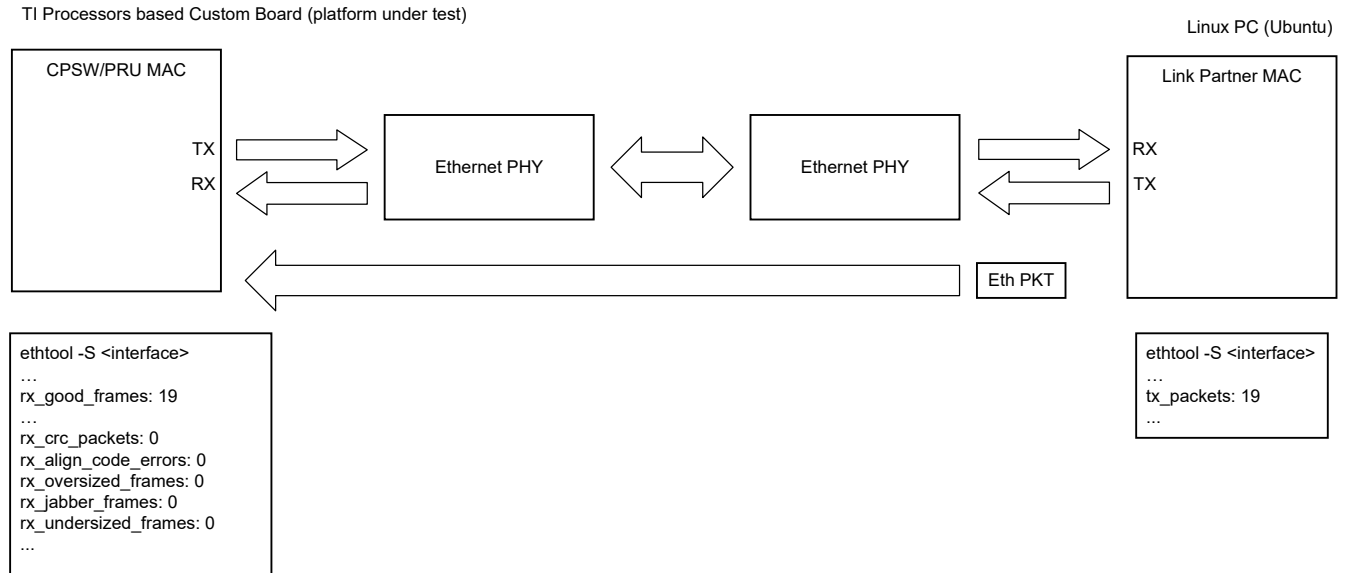


图 9-2. DUT 接收路径

当以太网 MAC 接收错误计数器不为零时，通常有两种可能的原因。最常见的原因是 PHY 接口模式为 RGMII 时的 RX 时钟时序不正确。第二常见的原因是硬件问题，例如不正确的内联电阻值或电路板布局布线问题。

### 备注

调试建议：

- 如果 PHY 接口模式为 RGMII，请在 DTS 文件中检查布局时序是否与 PHY 节点中编程的时序匹配。
- 参考 [高速接口布局指南](#) 应用手册以检查电路板布局布线。

如果分析 DUT 发送和接收路径的错误计数器为零，则表示基本以太网数据包的发送和接收没有错误。此时，可以添加 DHCP 服务器来为 DUT 动态分配 IP 地址，也可以在 DUT 上使用 `ip addr add <ip address/subnet mask> dev <interface name>` 来分配静态 IP 地址。但是，执行此步骤后，如果在检测到链路正常时仍然没有分配 IP 地址，并且以太网 MAC 统计信息没有显示任何错误，那么下一步是考虑可能存在网络原因。请查看 [节 10](#) 中的网络问题示例。

章节摘要：

- 本节介绍如何使用 FTP 方法分析基本的以太网数据包发送和接收。
- 如果数据包没有错误且未获得 IP 地址，则这可能意味着存在网络层面的问题。

## 10 调试网络拓扑和技术

本节中介绍的调试技术在不同网络问题下可能很有用。

### 10.1 直接连接

最简单的拓扑是将两个链路伙伴直接连接在一起，无需交换机和 WLAN 路由器。一个直接连接示例是 DUT 与已知良好的测试平台（例如 Ubuntu Linux PC）之间进行连接。图 10-1 显示了两个链路伙伴不使用交换机和 WLAN 路由器的直接连接拓扑。

连接两个链路伙伴后无需进一步操作即可生成流量，因为这两个平台都会发送广播数据包以尝试获取 IP 地址。

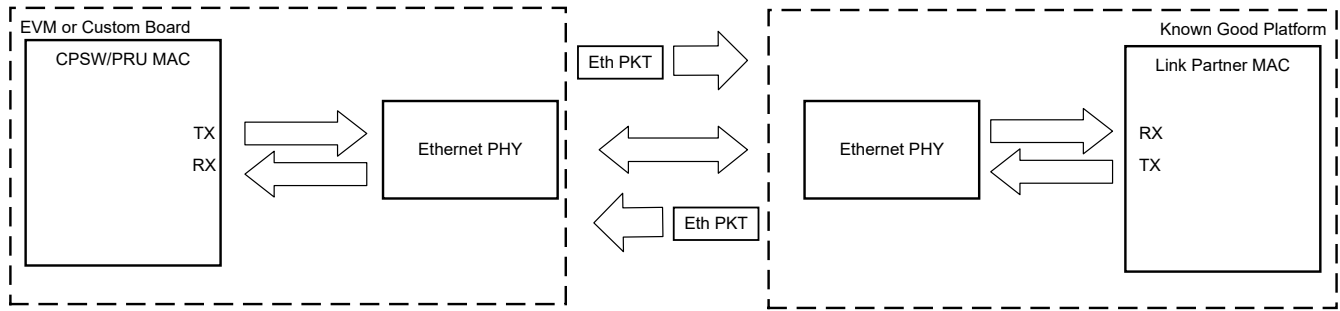


图 10-1. 直接连接拓扑

## 10.2 静态 IP 地址

要使用 `ping` 或 `iperf` 等实用工具启动通信，请设置 IPv4 地址。为此，可以使用 DHCP 服务器来分配动态 IP 地址，也可以手动分配静态 IP 地址。使用静态方法和直接连接拓扑时，两个链路伙伴需要手动分配到同一子网的 IP 地址。例如，如果 IPv4 地址为 192.168.1.1，网络掩码为 24，则前三组数字（即 192.168.1.x）为子网。节 11 提供了一个示例，说明如何在运行时设置静态 IP 地址。

## 10.3 设置低比特率

当使用 RGMII 接口模式时，可以通过降低以太网链路允许的最大比特率的方法来确定与数据有关的时钟延迟的相关时序问题。这一限制可通过三种方法实现：

1. 使用 `ethtool` 在两个链路伙伴上手动设置链路速度：  
`ethtool -s <interface name> speed <link speed>`
2. 连接到具有更低比特率的链路伙伴。
3. 在 DTS 中添加一个 `max-speed` 属性以降低比特率。此方法是推荐使用的方法，因为这是在引导期间在每个 DUT 上进行该设置。以下代码片段显示了在 DTS 文件中将此属性添加到 PHY 节点的示例。观察此处的速度设置为最低设置 10，表示 10Mbps。100Mbps 也是有效的设置。此处假设 PHY 硬件配置中的默认速度为 1000Mbps。

```
cpsw3g_phy1: ethernet-phy1@ {
    ...
    max-speed = <10>;
    ...
};
```

## 10.4 连接交换机时的注意事项

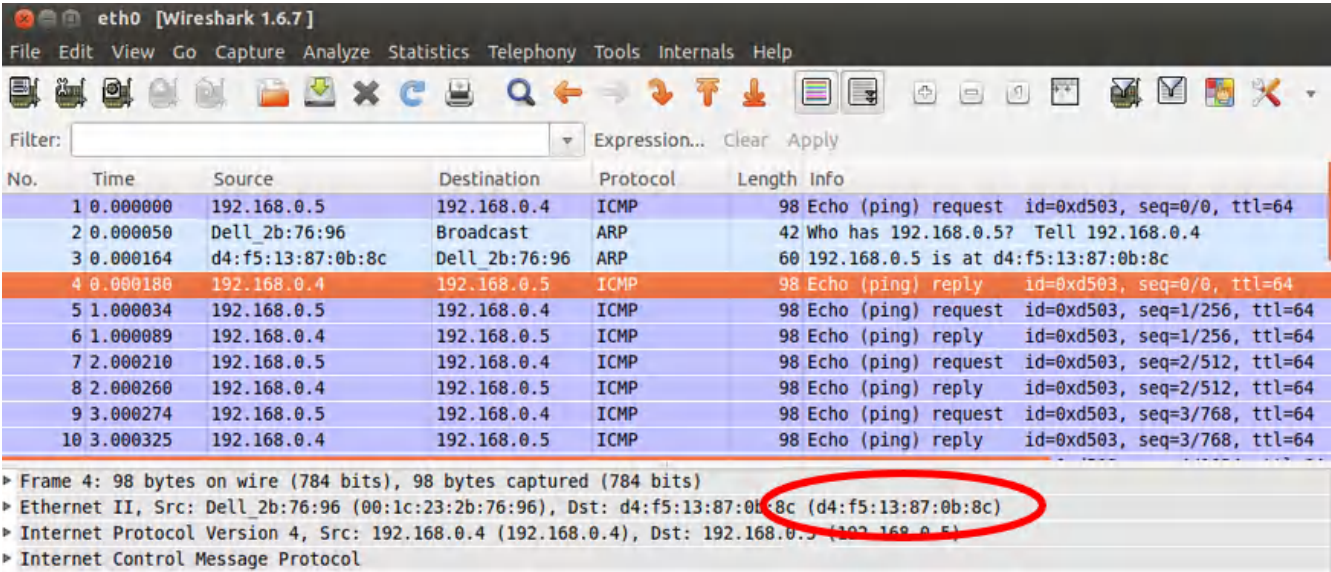
请勿将双端口板（例如 TI EVM）上的两个以太网接口连接到同一交换机或路由器。这种做法不会提供冗余链路。每个以太网接口必须有专用子网。

为了解释此网络拓扑失败的原因，假设 DUT 上的两个接口名为 `eth0` 和 `eth1`，并且都连接到同一交换机。从 `eth1` 发出 ping 命令（`ping <Ipv4 address> -I eth1`）（即发出 ICMP 请求）时，目标平台将接收由 DUT 发出的数据包。为了响应 ping 请求，目标平台必须执行地址解析协议（ARP）序列来获取发出 ping 请求的源平台的 MAC 地址。此 ARP 序列称为 ARP 请求消息。

由于 ARP 数据包是广播消息，源平台的 `eth0` 会在响应中提供 MAC 地址，这对于 `eth1` 来说是不正确的。

如果 ARP 广播返回了 `eth1` 的错误 MAC 地址，则来自目标平台的每个 ping 应答都会有错误的目标 MAC 地址。因此，接收所有响应的是 `eth0` 而不是 `eth1`。如果使用 Wireshark 来查看数据包流量，似乎每个 ping 请求都有匹配的 ping 应答，但 `eth1` 不会收到任何响应数据包，这会导致 ping 失败。

例如，在图 10-2 中，突出显示的 ping 应答是对目标 MAC 地址 `d4:f5:13:87:0b:8c` 的应答，该地址是与 `eth0` 关联的 MAC 地址。



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.5	192.168.0.4	ICMP	98	Echo (ping) request id=0xd503, seq=0/0, ttl=64
2	0.000050	Dell_2b:76:96	Broadcast	ARP	42	Who has 192.168.0.5? Tell 192.168.0.4
3	0.000164	d4:f5:13:87:0b:8c	Dell_2b:76:96	ARP	60	192.168.0.5 is at d4:f5:13:87:0b:8c
4	0.000180	192.168.0.4	192.168.0.5	ICMP	98	Echo (ping) reply id=0xd503, seq=0/0, ttl=64
5	1.000034	192.168.0.5	192.168.0.4	ICMP	98	Echo (ping) request id=0xd503, seq=1/256, ttl=64
6	1.000089	192.168.0.4	192.168.0.5	ICMP	98	Echo (ping) reply id=0xd503, seq=1/256, ttl=64
7	2.000210	192.168.0.5	192.168.0.4	ICMP	98	Echo (ping) request id=0xd503, seq=2/512, ttl=64
8	2.000260	192.168.0.4	192.168.0.5	ICMP	98	Echo (ping) reply id=0xd503, seq=2/512, ttl=64
9	3.000274	192.168.0.5	192.168.0.4	ICMP	98	Echo (ping) request id=0xd503, seq=3/768, ttl=64
10	3.000325	192.168.0.4	192.168.0.5	ICMP	98	Echo (ping) reply id=0xd503, seq=3/768, ttl=64

▶ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)  
 ▶ Ethernet II, Src: Dell\_2b:76:96 (00:1c:23:2b:76:96), Dst: d4:f5:13:87:0b:8c (d4:f5:13:87:0b:8c)  
 ▶ Internet Protocol Version 4, Src: 192.168.0.4 (192.168.0.4), Dst: 192.168.0.5 (192.168.0.5)  
 ▶ Internet Control Message Protocol

图 10-2. Wireshark Ping 消息



## 11 Linux® 实用程序汇总

本节介绍用于执行控制和状态操作的 Linux 实用程序工具。以太网接口调试中有三个常用的实用程序：**ifconfig**、**ip** 和 **ethtool**。以下是每个实用程序的简要说明。

- **ifconfig** - 提供有关接口的基本信息。输出中将汇总接口的当前状态。以下示例显示接口为 **up** 状态并具有 IP 地址 (192.168.1.5)。此外，还有用于表示 **RX** 和 **TX** 数据包数量的计数器。这里给出的示例显示接口正常工作。**ifconfig** 实用程序仅提供比 **ethtool** 更高网络层的统计数据，后者提供数据链路层结果。另外，**ifconfig** 比 **ip** 实用工具提供的信息更滞后。

```

root@am62xx-evm:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.5 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::1e63:49ff:fe1f:d9b2 prefixlen 64 scopeid 0x20<link>
    ether 1c:63:49:1f:d9:b2 txqueuelen 1000 (Ethernet)
    RX packets 552490 bytes 61792471 (58.9 MiB)
    RX errors 0 dropped 6 overruns 0 frame 0
    TX packets 646 bytes 50358 (49.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  
```

- **ip** - 用于控制和查询网络接口的网络工具。**ip** 实用程序比 **ifconfig** 实用程序的用途更广泛。
  - 以下示例说明了如何列出 **eth0** 接口的 IP 地址。

```

root@am62xx-evm:~# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 1c:63:49:1f:d9:b2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.5/24 metric 1024 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 73177sec preferred_lft 73177sec
    inet6 fe80::1e63:49ff:fe1f:d9b2/64 scope link
        valid_lft forever preferred_lft forever
  
```

- 以下示例说明了如何使用 **ip** 实用程序来设置静态 IP 地址。这种情况用于不存在 DHCP 服务器时的直接连接拓扑。

```

root@am62xxsip-evm:~# ip addr add 192.16.1.1/24 dev eth0
root@am62xxsip-evm:~#
root@am62xxsip-evm:~# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 08:04:b4:32:5d:ce brd ff:ff:ff:ff:ff:ff
    inet 192.16.1.1/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a04:b4ff:fe32:5dce/64 scope link
        valid_lft forever preferred_lft forever
  
```

- 以下示例显示了如何使用 **ip** 实用程序来检查 **RX** 和 **TX** 计数器。

```

root@am64xx-evm:~# ip -s link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
  default qlen 0
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX:  bytes  packets  errors  dropped  missed  mcast
         9246     99      0       0       0       0
    TX:  bytes  packets  errors  dropped  carrier  collsns
         9246     99      0       0       0       0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
  default ql 0
    link/ether 1c:63:49:1a:da:62 brd ff:ff:ff:ff:ff:ff
    RX:  bytes  packets  errors  dropped  missed  mcast
         19654     61      0       0       0       0
    TX:  bytes  packets  errors  dropped  carrier  collsns
         22919     88      0       0       0       0
...
  
```

- **ethtool** - 提供了一种方法来显示和修改以太网接口和相关器件驱动程序的参数。**ethtool** 有几个运行选项。两个最常用的选项是检查链路状态和 **MAC** 硬件统计信息。本应用手册全篇以示例的形式说明了这些具体选项。

## 12 申请以太网支持的检查清单

在 [TI E2E™ 论坛](#) 上发布咨询 ( 与在 DUT 上使用 Linux 操作系统时出现的以太网通信问题相关 ) 时, 请在初始帖子中回答以下问题或完成要求的步骤:

1. 使用的 TI SDK 版本是什么?
2. 使用的 TI 处理器是什么?
3. DUT 是 TI EVM 还是定制电路板?
4. 测试拓扑是什么?
5. 如果使用定制电路板, 附加定制电路板的 DTS 文件; 不要剪切并粘贴 DTS 内容, 因为 DTS 文件很长, 读者很难解析
6. 附加 DUT 的控制台引导日志
7. 附加有问题的接口的 **ethtool <interface name>** 和 **ethtool -S <interface name>** 结果
8. 指明您已查看本应用手册

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024，德州仪器 (TI) 公司