UDP 通信研究报告

本文通过对以太网通信中的 UDP 传输协议的理论学习，针对 UDP 实际应用中的丢包问题，提出一种人为的重发机制完成 UDP 稳定可靠的传输，并通过实验进行了验证。

1 以太网简介

以太网是一种产生较早，使用相当广泛的局域网。其最初是由 Xerox（施乐）公司创建并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范，后来被电气与电子工程师协会（IEEE）所采纳作为 802.3 的标准。

以太网的分类有标准以太网（10Mbit/s），快速以太网(100Mbit/s)和千兆以太网（1000Mbit/s）。随着以太网技术的飞速发展，市场上也出现了万兆以太网（10Gbit/s），它扩展了 IEEE802.3 协议和 MAC 规范，使其技术支持 10Gbit/s 的传输速率。然而在实际应用中，标准以太网和快速以太网已经能够满足我们的日常需求，对通信速率要求较高的场合，才会用到千兆以太网。

以太网通信离不开连接端口的支持，网络数据连接的端口就是以太网接口。以太网接口类型有 RJ45 接口，RJ11 接口（电话线接口），SC 光纤接口等。其中 RJ45 接口是我们现在最常见的网络设备接口（如：电脑网口），我们开发板使用的就是这种接口。

RJ45 接口俗称“水晶头”，专业术语为 RJ45 连接器，由插头（接头、水晶头）和插座（母座）组成，属于双绞线以太网接口类型。RJ45 插头只能沿固定方向插入，设有一个塑料弹片与 RJ45 插槽卡住以防止脱落，如图 1.1 所示：

图 1.1RJ45 插头（左）、插座（右）

以太网是目前应用最广泛的局域网通讯方式，同时也是一种协议。以太网协议定义了一系列软件和硬件标准，从而将不同的计算机设备连接在一起。我们知道串口通信单次只传输一个字节，而以太网通信是以数据包的形式传输，其单包数据量达到几十，甚至成百上千个字节。

2UDP 协议

图 2.1 为以太网通过 UDP（UserDatagramProtocol，用户数据报协议）传输单包数据的格式，从图中可以看出，以太网的数据包就是对各层协议的逐层封装来实现数据的传输。

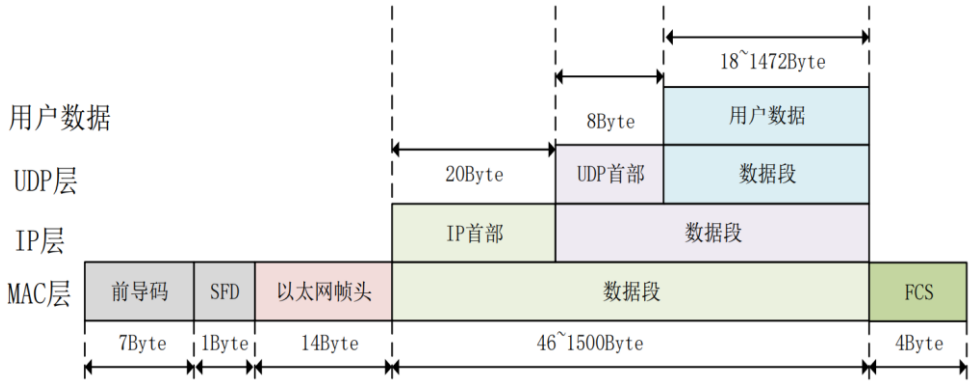

图 2.1 以太网包数据格式

2.1 以太网 MAC 帧格式

以太网技术的正式标准是 IEEE802.3，它规定了以太网传输数据的帧结构，我们可以把以太网 MAC 层理解成高速公路，我们必须遵循它的规则才能在上面通行，以太网 MAC 层帧格式如图 2.2 所示。

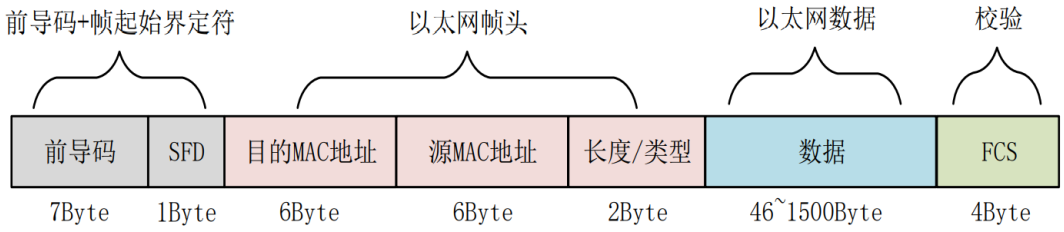

图 2.2 以太网 MAC 帧格式

以太网传输数据时按照上面的顺序从头到尾依次被发送和接收，我们下面进一步解释各个区域。

（1）前导码（Preamble）：为了实现底层数据的正确阐述，物理层使用 7 个字节同步码（0 和 1 交替（0x55-0x55-0x55-0x55-0x55-0x55-0x55））实现数据的同步。

（2）帧起始界定符（SFD，StartFrameDelimiter）：使用 1 个字节的 SFD（固定值为 0xd5）来表示一帧的开始，即后面紧跟着传输的就是以太网的帧头。

（3）目的 MAC 地址：即接收端物理 MAC 地址，占用 6 个字节。MAC 地址从应用上可分为单播地址、组播地址和广播地址。单播地址：第一个字节的最

低位为 0，比如 00-00-00-11-11-11，一般用于标志唯一的设备；组播地址：第一个字节的最低位为 1，比如 01-00-00-11-11-11，一般用于标志同属一组的多个设备；广播地址：所有 48bit 全为 1，即 FF-FF-FF-FF-FF-FF，它用于标志同一网段中的所有设备。

（4）源 MAC 地址：即发送端物理 MAC 地址，占用 6 个字节。

（5）长度/类型：图 2.2 中的长度/类型具有两个意义，当这两个字节的值小于 1536（十六进制为 0x0600）时，代表该以太网中数据段的长度；如果这两个字节的值大于 1536，则表示该以太网中的数据属于哪个上层协议，例如 0x0800 代表 IP 协议（网际协议）、0x0806 代表 ARP 协议（地址解析协议）等。

（6）数据：以太网中的数据段长度最小 46 个字节，最大 1500 个字节。最大值 1500 称为以太网的最大传输单元（MTU，MaximumTransmissionUnit），之所以限制最大传输单元是因为在多个计算机的数据帧排队等待传输时，如果某个数据帧太大的话，那么其它数据帧等待的时间就会加长，导致体验变差，这就像一个十字路口的红绿灯，你可以让绿灯持续亮一小时，但是等红灯的人一定不愿意的。另外还要考虑网络 I/O 控制器缓存区资源以及网络最大的承载能力等因素，因此最大传输单元是由各种综合因素决定的。为了避免增加额外的配置，通常以太网的有效数据字段小于 1500 个字节。

（7）帧检验序列（FCS，FrameCheckSequence）：为了确保数据的正确传输，在数据的尾部加入了 4 个字节的循环冗余校验码（CRC 校验）来检测数据是否传输错误。CRC 数据校验从以太网帧头开始即不包含前导码和帧起始界定符。通用的 CRC 标准有 CRC-8、CRC-16、CRC-32、CRC-CCIT，其中在网络通信系统中应用最广泛的是 CRC-32 标准。

在这里还有一个要注意的地方就是以太网相邻两帧之间的时间间隔，即帧间隙（IFG，InterpacketGap）。帧间隙的时间就是网络设备和组件在接收一帧之后，需要短暂的时间来恢复并为接收下一帧做准备的时间，IFG 的最小值是 96bit time，即在媒介中发送 96 位原始数据所需要的时间，在不同媒介中 IFG 的最小值是不一样的。不管 10M/100M/1000M 的以太网，两帧之间最少要有 96bit time，IFG 的最少间隔时间计算方法如下：

（1）10Mbit/s 最小时间为：96*100ns=9600ns；

（2）100Mbit/s 最小时间为：96*10ns=960ns；

（3）1000Mbit/s 最小时间为：96*1ns=96ns。

2.2IP 协议

IP 协议是 TCP/IP 协议簇中的核心协议，也是 TCP/IP 协议的载体，IP 协议规定了数据传输时的基本单元和格式。从前面介绍的图 43.1.3 中可以看出，IP 协议位于以太网 MAC 帧格式的数据段，IP 协议内容由 IP 首部和数据字段组成。所有的 TCP、UDP 及 ICMP 数据都以 IP 数据报格式传输，IP 数据包格式如图 2.3 所示。

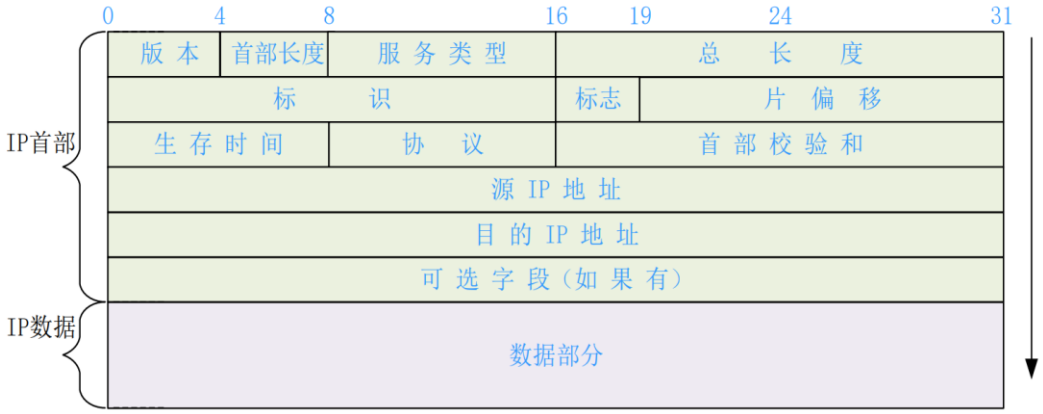

图 2.3IP 数据报格式

前 20 个字节和紧跟其后的可选字段是 IP 数据报的首部，前 20 个字节是固定的，后面可选字段是可有可无的，首部的每一行以 32 位（4 个字节）为单位。

（1）版本：4 位 IP 版本号（Version），这个值设置为二进制的 0100 时表示 IPv4，设置为 0110 时表示 IPv6，目前使用比较多的 IP 协议版本号是 4。

（2）首部长度：4 位首部长度（IHL，InternetHeaderLength），表示 IP 首部一共有多少个 32 位（4 个字节）。在没有可选字段时，IP 首部长度为 20 个字节，因此首部长度的值为 5。

（3）服务类型：8 位服务类型（TOS，Typeofservice），该字段被划分成两个子字段：3 位优先级字段（现在已经基本忽略掉了）和 4 位 TOS 字段，最后一位固定为 0。服务类型为 0 时表示一般服务。

（4）总长度：16 位 IP 数据报总长度（TotalLength），包括 IP 首部和 IP 数据部分，以字节为单位。我们利用 IP 首部长度和 IP 数据报总长度，就可以知道 IP 数据报中数据内容的起始位置和长度。由于该字段长 16bit，所以 IP 数据报最长可达 65535 字节。尽管理论上可以传输长达 65535 字节的 IP 数据报，但实际上还要考虑网络的最大承载能力等因素。

（5）标识字段：16 位标识（Identification）字段，用来标识主机发送的每一份数据报。通常每发送一份报文它的值就会加 1。

（6）标志字段：3 位标志（Flags）字段，第 1 位为保留位；第 2 位表示禁止分片（1 表示不分片 0：允许分片）；第 3 位标识更多分片（除了数据报的最后一个分片外，其它分片都为 1）。

（7）片偏移：13 位片偏移（FragmentOffset），在接收方进行数据报重组时用来标识分片的顺序。

（8）生存时间：8 位生存时间字段，TTL（TimeToLive）域防止丢失的数据包在无休止的传播，一般被设置为 64 或者 128。

（9）协议：8 位协议（Protocol）类型，表示此数据报所携带上层数据使用的协议类型，ICMP 为 1，TCP 为 6，UDP 为 17。

（10）首部校验和：16 位首部校验和（HeaderChecksum），该字段只校验数据报的首部，不包含数据部分；校验 IP 数据报头部是否被破坏、篡改和丢失等。

（11）目的 IP 地址：32 位目的 IP 地址（DestinationAddress），即接收端的 IP 地址，如 192.168.0.3。

（12）可选字段：是数据报中的一个可变长度的可选信息，选项字段以 32bit 为界，不足时插入值为 0 的填充字节，保证 IP 首部始终是 32bit 的整数倍。

以上内容是对 IP 首部格式的详细阐述，还需要补充的内容是 IP 首部校验和的计算方法，其计算步骤如下：

1、将 16 位检验和字段置为 0，然后将 IP 首部按照 16 位分成多个单元；

2、对各个单元采用反码加法运算（即高位溢出位会加到低位，通常的补码运算是直接丢掉溢出的高位）；

3、此时仍然可能出现进位的情况，将得到的和再次分成高 16 位和低 16 位进行累加；

4、最后将得到的和的反码填入校验和字段。

例如，我们使用 IP 协议发送一个 IP 数据报总长度为 50 个字节（有效数据为 30 个字节）的数据包，发送端 IP 地址为 192.168.1.123，接收端 IP 地址为 192.168.1.102，则 IP 首部数据如图 2.4 下：

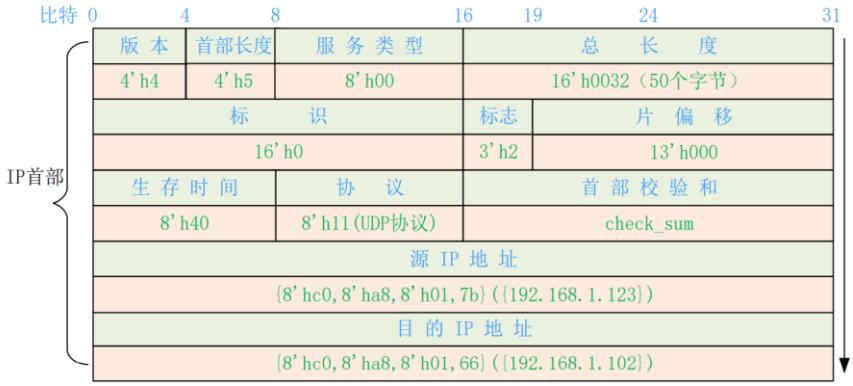

图 2.4IP 首部数据

按照上述提到的 IP 首部校验和的方法计算 IP 首部校验和，即：

1、0x4500+0x0032+0x0000+0x4000+0x4011+0x0000（计算时强制置 0）
+0xc0a8+0x017b+0xc0a8+0x0166=0x24974

2、0x0002+0x4974=0x4976

3、0x0000+0x4976=0x4976（此种情况并未出现进位）

4、check_sum=~0x4976（按位取反）=0xb689

2.3UDP 协议

事实上数据是可以直接封装在 IP 协议里而不使用 TCP、UDP 或者其它上层协议的。然而在网络传输中同一 IP 服务器需要提供各种不同的服务，各种不同的服务类型是使用端口号来区分的，例如用于浏览网页服务的 80 端口，用于 FTP（文件传输协议）服务的 21 端口等。TCP 和 UDP 都使用两个字节的端口号，理论上可以表示的范围为 0~65535，足够满足各种不同的服务类型。

然后是为什么不选择传输更可靠的 TCP 协议，而是 UDP 协议呢？TCP 协议与 UDP 协议作为传输层最常用的两种传输协议，这两种协议都是使用 IP 作为网络层协议进行传输。下面是 TCP 协议与 UDP 协议的区别：

（1）TCP 协议面向连接，是流传输协议，通过连接发送数据，而 UDP 协议传输不需要连接，是数据报协议；

（2）TCP 为可靠传输协议，而 UDP 为不可靠传输协议。即 TCP 协议可以保证数据的完整和有序，而 UDP 不能保证；

（3）UDP 由于不需要连接，故传输速度比 TCP 快，且占用资源比 TCP 少；

（4）应用场合：TCP 协议常用在对数据文件完整性较高的一些场景中，如文件传输等。UDP 常用于对通讯速度有较高要求或者传输数据较少时，比如对速度要求较高的视频直播和传输数据较少的 QQ 等。

首先使用 FPGA 实现 TCP 协议是完全没有问题的，但是，FPGA 发展到现在，却鲜有成功商用的 RTL 级的 TCP 协议设计，大部分以太网传输都是基于比较简单的 UDP 协议。TCP 协议设计之初是根据软件灵活性设计的，如果使用硬件逻辑实现，工程量会十分巨大，而且功能和性能无法得到保证，因此，TCP 协议设计并不适合使用硬件逻辑实现。UDP 协议是一种不可靠传输，发送方只负责数据发送出去，而不管接收方是否正确的接收。在很多场合，是可以接受这种潜在的不可靠性的，例如视频实时传输显示等。

UDP 数据格式如图 2.5 所示：

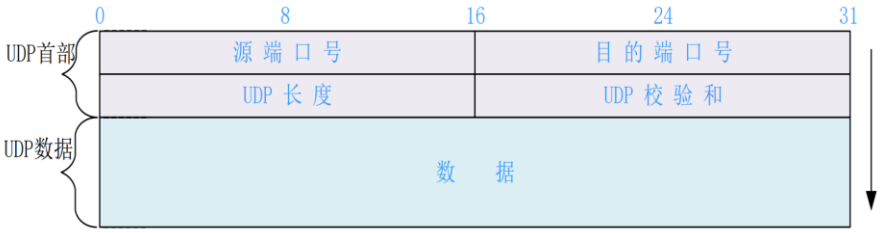

图 2.5UDP 数据格式

UDP 首部共 8 个字节，同 IP 首部一样，也是一行以 32 位（4 个字节）为单位。

（1）源端口号：16 位发送端端口号，用于区分不同服务的端口，端口号的范围从 0 到 65535。

（2）目的端口号：16 位接收端端口号。

（3）UDP 长度：16 位 UDP 长度，包含 UDP 首部长度+数据长度，单位是字节（byte）。

（4）UDP 校验和：16 位 UDP 校验和。UDP 计算校验和的方法和计算 IP 数据报首部校验和的方法相似，但不同的是 IP 数据报的校验和只检验 IP 数据报的首部，而 UDP 校验和包含三个部分：UDP 伪首部，UDP 首部和 UDP 的数据部分。伪首部的数据是从 IP 数据报头和 UDP 数据报头获取的，包括源 IP 地址，目的 IP 地址，协议类型和 UDP 长度，其目的是让 UDP 两次检查数据是否已经正确到达目的地，只是单纯为了做校验用的。在大多数使用场景中接收端并不检测 UDP 校验和，因此这里不做过多介绍。

2.4 以太网 PHY 芯片

PHY 在发送数据的时候，接收 MAC 发过来的数据（对 PHY 来说，没有帧的概念，都是数据而不管什么地址，数据还是 CRC），把并行数据转化为串行流数据，按照物理层的编码规则把数据编码转换为模拟信号发送出去，收数据时的流程反之。PHY 还提供了和对端设备连接的重要功能并通过 LED 灯显示出自己目前的连接状态和工作状态。当我们给网卡接入网线的时候，PHY 芯片不断发出脉冲信号来检测对端是否有设备，它们通过标准的“语言”交流，互相协商并确定连接速度、双工模式、是否采用流控等。通常情况下，协商的结果是两个设备中能同时支持的最大速度和最好的双工模式。这个技术被称为 AutoNegotiation，即自动协商。以太网 MAC 和 PHY 之间有一个接口，常用的接口有 MII 和 GMII 等。

（1）MII（MediumIndependentInterface）：MII 支持 10Mbps 和 100Mbps 的操作，数据位宽为 4 位，在 100Mbps 传输速率下，时钟频率为 25Mhz。

（2）GMII（GigabitMII）：GMII 接口向下兼容 MII 接口，支持 10Mbps、100Mbps 和 1000Mbps 的操作，数据位宽为 8 位，在 1000Mbps 传输速率下，时钟频率为 125Mhz。

3UDP 通信实验

3.1 硬件连接

图 3.1 为 FPGA 与以太网 PHY 芯片连接示意图

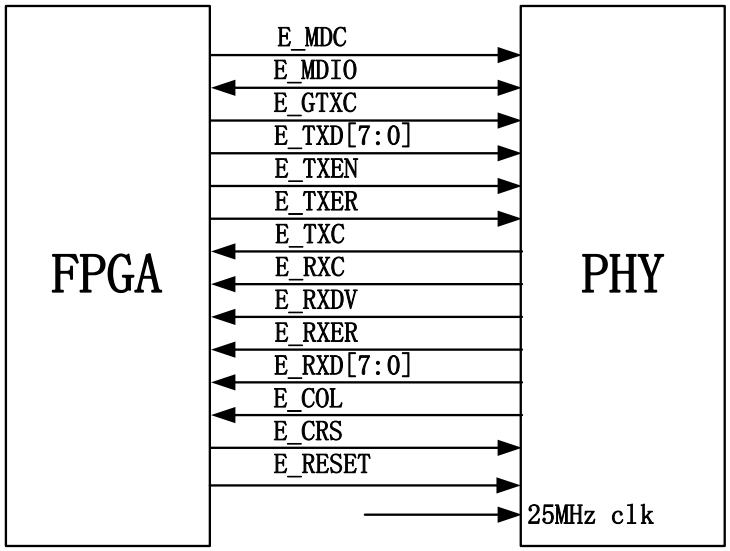

图 3.1FPGA 与 PHY 连接示意图

图 3.1 中各引脚说明如表 3.1 所示。

表 3.1PHY 引脚含义

引脚名称	说明	引脚名称	说明
E_MDC	MDIO 管理时钟	E_CRS	Carrier Sense 信号
E_MDIO	MDIO 管理数据	E_COL	Collosion 信号
E_GTXC	GMII 发送时钟	E_RESET	复位信号
E_TXD[7:0]	8bit 并行发送数据线	E_RXD[7:0]	8bit 并行接收数据线
E_TXEN	发送使能信号	E_RXDV	接收数据有效信号
E_TXER	发送错误新年好	E_RXER	接收数据错误信号
E_TXC	MII 发送时钟	E_RXC	GMII 接收时钟
25MHz clk	PHY 外部输入时钟		

注：PHY 外部输入时钟一般由晶振提供，当没有晶振时可由 FPGA 提供

3.2 程序设计

通过前面介绍的以太网相关协议和 GMII 接口可知，我们只需要把数据封装成以太网包的格式通过 GMII 接口传输数据即可。整个 UDP 通信工程可分为接收模块、发送模块以及 CRC 校验模块。

3.2.1 接收模块

以太网的接收模块较为简单，因为我们不需要对数据做 IP 首部校验也不需要做 CRC 循环冗余校验，只需要判断目的 MAC 地址与开发板 MAC 地址、目的 IP 地址与开发板 IP 地址是否一致即可。接收模块的解析顺序是：前导码+帧起始界定符→以太网帧头→IP 首部→UDP 首部→UDP 数据（有效数据）→接收结束。GMII 接口数据为 8 位数据，IP 数据报一般以 32bit 为单位，为了和 IP 数据报格式保持一致，所以要把 8 位数据转成 32 位数据，因此接收模块实际上是完成了 8 位数据转 32 位数据的功能。

整个接收模块的状态跳转图如图 3.2 所示

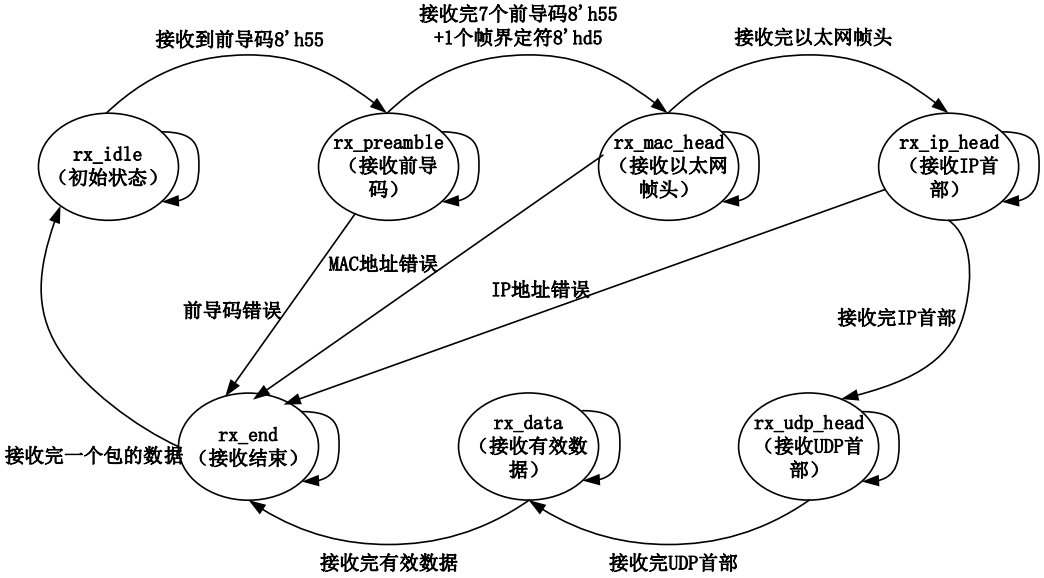

图 3.2 接收模块的状态跳转图

接收模块使用三段式状态机来解析以太网包，从上图可以比较直观的看到每个状态实现的功能以及跳转到下一个状态的条件。这里需要注意的一点是，在中间状态如前导码错误、MAC 地址错误以及 IP 地址错误时跳转到 rx_end 状态而不是跳转到 rx_idle 转态。因为中间状态在解析到数据错误时，单包数据的接收还没有结束，如果此时跳转到 rx_idle 状态会误把有效数据当成前导码来解析，所以状态跳转到 rx_end。

3.2.2 发送模块

以太网发送模块和接收模块比较类似，但是多了 IP 首部校 验和和 CRC 循环冗余校验的计算。CRC 的校验并不是在发送模块完成，而是在 CRC 校验模块（crc32_d8）里完成的。发送模块的发送顺序是前导码+帧起始界定符→以太网帧头→IP 首部 →UDP 首部→UDP 数据（有效数据）→CRC 校验。输入的有效数据为 32 位数据，GMII 接口为 8 位数 据接口，因此发送模块实际上完成的是 32 位数据转 8 位数据的功能。

整个流程状态机如图 3.3 所示

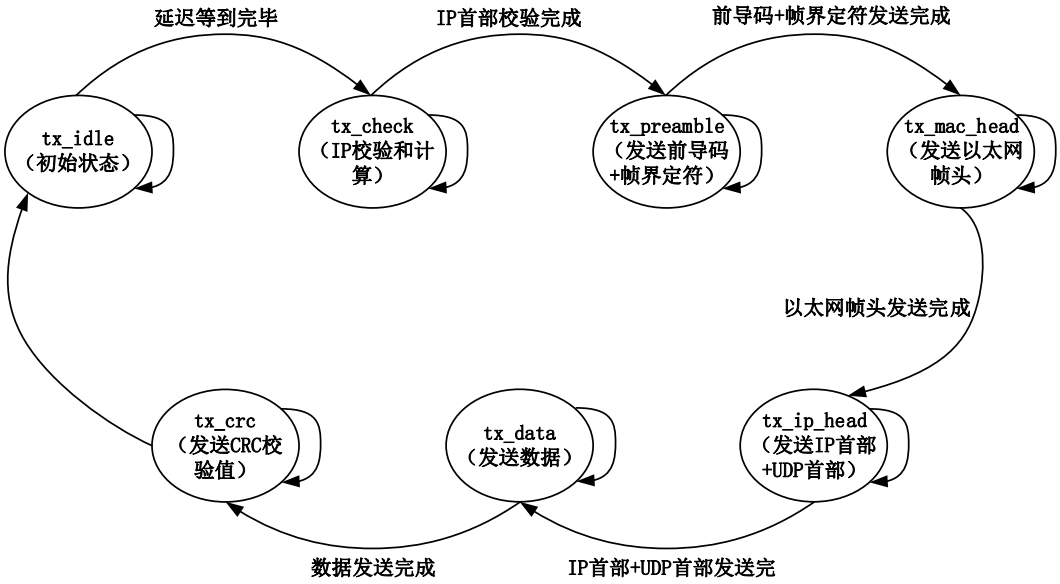

图 3.3 发送模块的状态跳转图

发送模块和接收模块有很多相似之处，同样使用三段式状态机来发送以太网包，从图 3.3 可以比较直观的看到每个状态实现的功能以及跳转到下一个状态的条件。其中发送模块以数组存储以太网的帧头、IP 首部以及 UDP 的首部信息，在复位时完成初始化。

3.2.3CRC 校验模块

CRC 校验模块是对以太网发送模块的数据（不包括前导码和帧起始界定符）做校验，把校验结果值拼在以太网帧格式的 FCS 字段，如果 CRC 校验值计算错误或 者没有的话，那么电脑网卡会直接丢弃该帧导致收不到数据（有些网卡是可以设置不做校验的）。 CRC32 校验在 FPGA 实现的原理是 LFSR（Linear Feedback Shift Register，线性反馈移位寄存 器），其思想是各个寄存器储存着上一次 CRC32 运算的结果，寄存器的输出即为 CRC32 的值。

一个 IP 数据包的 CRC32 校验是在目标 MAC Address 开始计算的，一直计算到一个包的 最后一个数据为止。以太网的 CRC32 的 verilog 的算法和多

项式可以在以下网站中直接生成：CRC Generation Tool - easics；参数设置如图 3.4 所示。

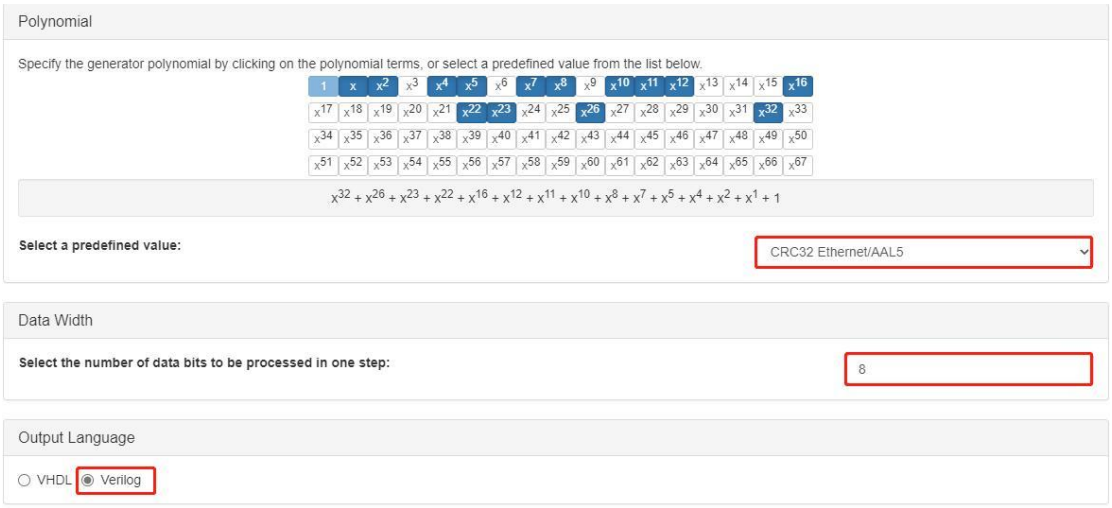

图 3.4CRC 校验参数设置

3.3 实验验证

3.3.1 通信参数设置

首先通过千兆网线连接电脑和 FPGA 电路板建立物理连接，接着进入计算机的网络设置“控制面板\网络和 Internet\网络连接”，选择以太网进行网络 IP 设置，具体操作如图 3.5 所示。

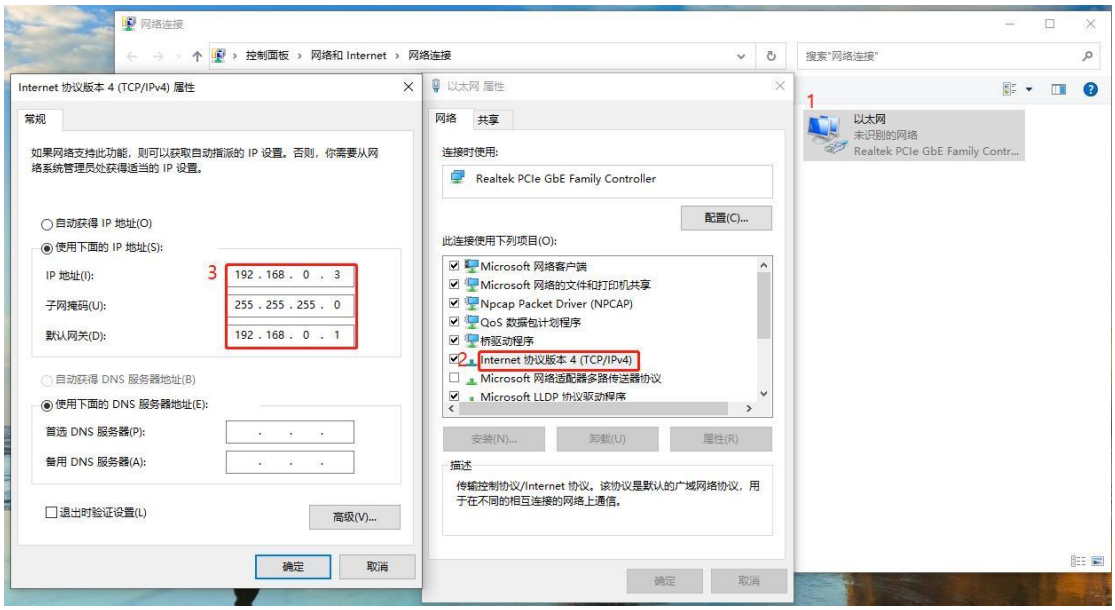

图 3.5 以太网 IP 设置流程

在完成 IP 设置后还需进行开发板的 IP 地址和 MAC 地址 DE 绑定；在开始界面以管理员身份运行命令提示符，如图 3.6 所示。

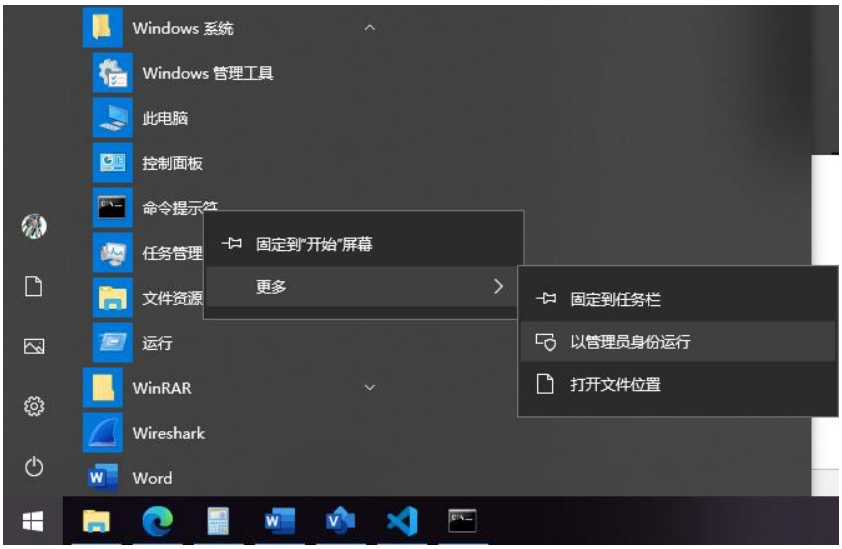

图 3.6 运行命令提示符

在弹出的对话框内输入 netsh i i show in 查看以太网的 id 地址，再输入 netsh -c i i add neighbors 2 192.168.0.2 00-0a-35-01-fe-c0，完成绑定；其中“2”为以太网的 id 号，“192.168.0.2”为 FPGA 板卡的 IP 地址，“00-0a-35-01-fe-c0”为 FPGA 板卡的 MAC 地址。

3.3.2UDP 通信测试

通过 3.3.1 节完成 IP 设置以及 ARP 绑定后，将代码下载至电路板中，此时 FPGA 会向 PC 端发送“Hello，Welcome to FPGA!”，当 PC 端向 FPGA 发送数据时，发送的数据会存入 FPGA 内部，之后 FPGA 会一直发送 PC 端发送的数据段，结果如图 3.7 所示。

图 3.7UDP 通信测试

注意：以太网的数据帧的传输有包长的要求，一般在 46～1500 字节。所以在发送以太网数据包的时候，数据帧的长度不能太短，不然会导致 PC 数据包发送而 FPGA 收不到数据包的情况。

3.3.3 自研重发机制下的 UDP 通信测试

针对 UDP 协议通信过程中会存在丢包的问题，通过对每个包作标记建立人为的重发机制，实现 UDP 的稳定可靠传输。

重发机制主模块主要由 FIFO 模块、RAM 模块、UDP 通信模块三部分组成。其中 FIFO 模块用于存放前端数据源；RAM 模块主要用于存放即将发送的 UDP 包的包序号和数据；UDP 模块负责发送数据和接收是否重发的指令；主模块负责根据接收到的指令来判断是否更新 RAM 中的数据，如果不需要重发，则通过将 FIFO 数据写入 RAM 中完成数据更新，反之，就不更新，整个重发机制流程图如图 3.8 所示。

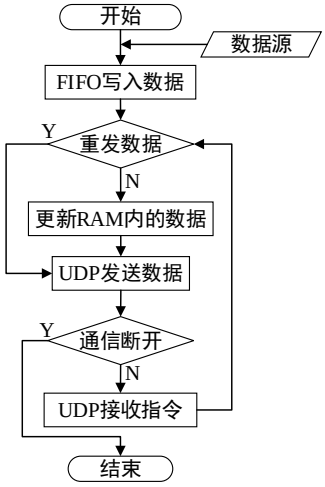

图 3.8 重发机制流程图

将采集到的连续累加数（累加 1）进行前后作差，如图 3.9 所示。

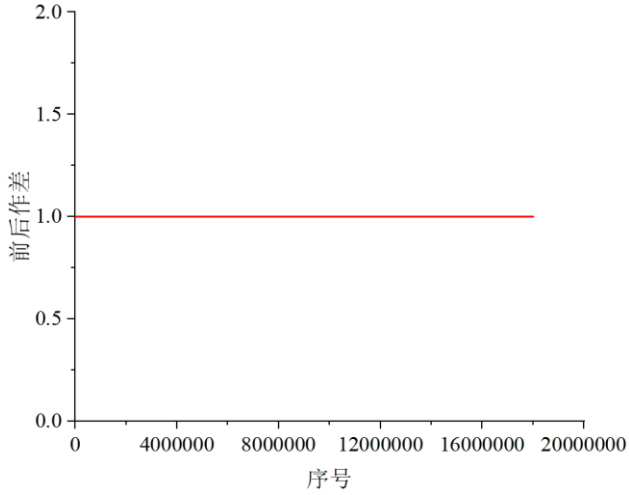

图 3.9 重发模块接收数据前后作差曲线

由图可以看出，无明显丢包现象。但是，在实际测试过程，上位机出现传输不稳定现象，需要在打开第三方抓包软件的帮助下才能保持数据传输稳定。

通过对抓包软件的原理分析，在自主编写的上位机中加入基于 wincap 的网卡监听功能，在摆脱第三方软件的情况下也能实现数据的稳定传输。

4 结论

目前，针对 UDP 丢包问题提出的重发机制方案可实现 2MB/s 的稳定可靠传输。但是该方案并不能从根本上解决 UDP 协议内的丢包现象，为了防止因为丢包频繁出现时，超时等待时间过长导致 FPGA 数据缓冲区被塞满引起丢数，等待时间需要设置很短。而又因为上位机的编程环境处于 TCP/IP 网络模型的应用层，FPGA 上传的数据需要经过几层的协议解析才会到达应用层，解析需要时间，因此超时等待时间不能设置太短，因此最好使用大容量的缓存作为 FPGA 数据缓冲区，比如 SDRAM。