**第二届“通达杯”ADI软件无线电大赛**

**项目报告**

团队名称： Misuzu

所在学校： 华中科技大学

所在学院： 电子信息与通信学院

队长姓名： 龚秋虎

队长邮箱： air\_misuzu@163.com

队员姓名： 李嘉淳、张雄

指导教师： 黑晓军

**小型组网通信系统**

摘要：本系统以PlutoSDR为平台，通过Matlab软件，设计并实现了一个小型组网通信系统。该系统采用BPSK调制方案，BPSK的传输码率为1Mbit/s。该系统使用三个子频段，设备可通过设置来配置自己的工作频段和接收频段，通过频分复用实现组网。发射端将待发送的文件拆分成多段数据，并将数据打包，加上包头和包尾的CRC校验发送出去。接收端收下属于自己的包并将数据组装成接收文件。该系统经过测试，目前可在两米以上的通信距离实现三台设备之间的任意组网以及两台设备的相互通信和单一设备的自发自收测试，支持包括txt和wav在内的多种格式的数据传输，数据误码率远小于1%，并提供良好的人机交互功能。

关键词：BPSK；组网；频分复用

# 1．系统方案

本系统可看作一个简单的三层协议系统，主要分为物理层，传输层和应用层。

该系统的物理层包括了OSI七层协议中物理层和部分链路层的功能。该层将传输层提交的数据帧转换成比特流之后进行BPSK调制发射出去，接收端则对接收到的信号进行BPSK解调，并进行帧定界和CRC校验，对于CRC校验失败的帧直接丢弃。该层还提供了频分组网功能，对于不同发射频段的设备，会将BPSK先调制到不同的频段，然后再发送出去。相应的，接收端则会先对想接收的频段进行带通滤波之后得到该频段信号再下变频到基频之后进行BPSK的IQ解调。

传输层提供端到端的数据传输，根据应用层提供的数据包将数据拆分成不同的数据报并组装成帧交付到物理层并在接收端将接收到的数据帧提取数据报重新组装成完整的数据包提交给应用层。传输层的服务需要经历连接建立阶段，数据传送阶段和连接释放阶段来完成一个完整的服务，以此来向上层提供可靠的数据传输。

应用层提供上层应用，包括人机交互和文件发送接收功能。用户可以通过人机交互功能来实现物理层和数据链路层的参数配置，以及配置发送接收频点。该系统的每个设备均提供两个通道，两个通道可任意配置成发送模式或者接收模式，或者是不工作。文件发送接收功能能够将自动读取文件并进行格式转换之后得到数据提供给传输层。接收端则接收传输层提交上来的数据，并重新转换成完整的文件给用户。

# 2．物理层设计

## 2.1 BPSK调制解调

二进制相移键控(BPSK)利用载波的相位变化来传递数字信息，而振幅和频率保持不变，具有良好的抗干扰能力。

发送端接收上层提供的数据帧，对其进行32位的CRC校验，并将CRC校验的结果添加到数据帧尾部得到完整的数据帧。为了保证传输的比特流0和1各占一半，需要先对数据帧用七位PN码进行加扰，得到伪随机的比特流。然后再进行BPSK调制。BPSK调制采用IQ调制方案，根据星座图将比特转换成对应的IQ信号。为了保证发射端和接收端的频率能够同步，在IQ信号中隔一段距离插入一段导频。最后在该IQ信号的前面插入由7位m序列生成的127比特同步序列即可得到完整的调制数据。此时，数据的码率为40Mbit/s。该数据目前还不适合直接放到信道中传输，因为其频谱分布太宽，因此需要对其再进行一次成型滤波并进行4倍抽取来限制其频谱，并降低码率至10Mbit/s。发送端框图如图2.1。

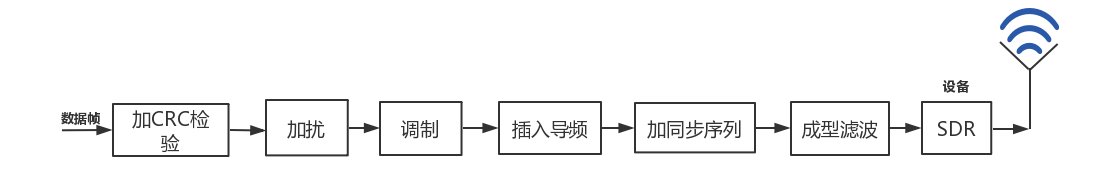


图2.1发送端框图

BPSK的接收端将接收的信号先进行匹配滤波，抵消发射端进行的成型滤波，并进行4倍插值来进行采样同步。为了便于信号的处理，接收端还会对数据进行归一化处理，以降低信道衰减对后续处理的影响。求接收到的数据和前导同步序列的相关性，确定最大相关位置，该位置即是同步序列的位置，之后即可对帧进行定界。进行帧定界之后提取帧数据进行后续处理，根据帧里面的导频信号进行频率和相位的校正，提高解调的性能。同步之后删除调导频序列并进行时域均衡之后，得到数据帧信号。此时的数据帧信号还是加扰过的，再解扰之后就得到了完整的数据帧。针对该数据帧进行CRC校验，校验成功的去除CRC尾部后提交给上层，校验失败的直接丢弃。接收端框图如图2.2

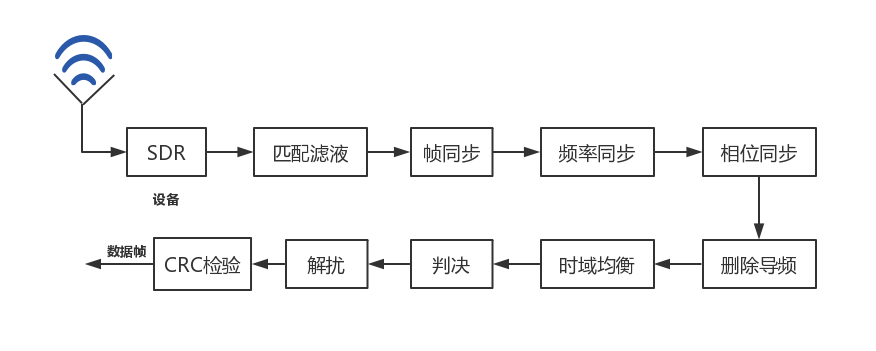


图2.2 接收端框图

## 2.2频分复用

该系统的PlutoSDR提供了20M的信号处理带宽，将该频带分为3个子频带用于频分复用。对于每一台设备，都有一个自己的发送频段和接收频段，不同的设备必须工作在不同的发送频段以避免信号干扰，但是可以工作在任意的接收频段接收数据。

数据经过BPSK调制之后的码率是10Mbit/s，带宽过宽，不方便直接进行频分复用。因此先进行10倍插值之后，将码率变换为1Mbit/s，然后再将其通过40M采样率下的归一化截至频率为0.1的低通滤波器滤除插值产生的多余频谱分量，保证信号的带宽限制在1M以内。之后将该信号调制到发送频段。本系统共提供三个工作频段，分别为2MHZ，5MHZ和8MHZ，每个频段间保留有3MHZ的间隔，以避免频带间干扰。

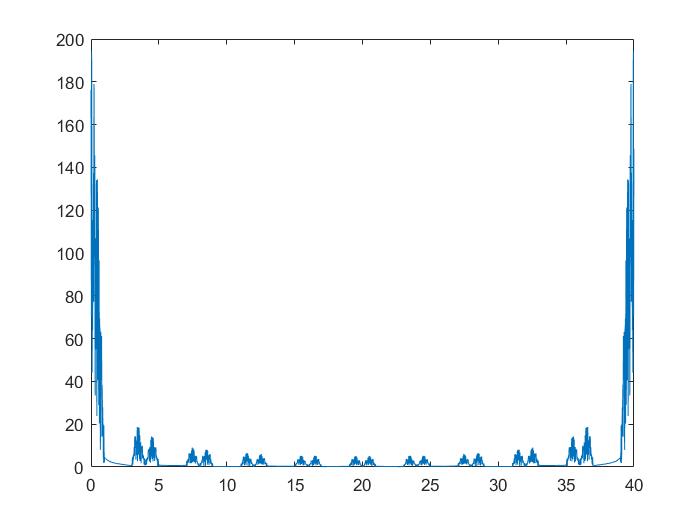
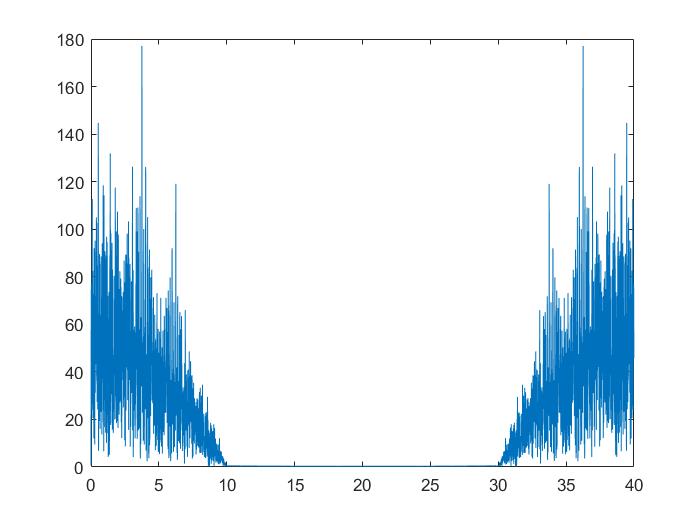


图2.3 BPSK调制信号频谱 图2.4 10倍插值后频谱

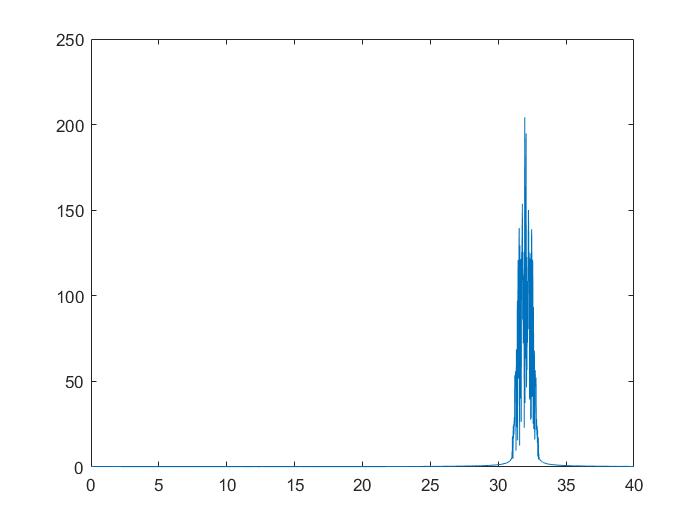
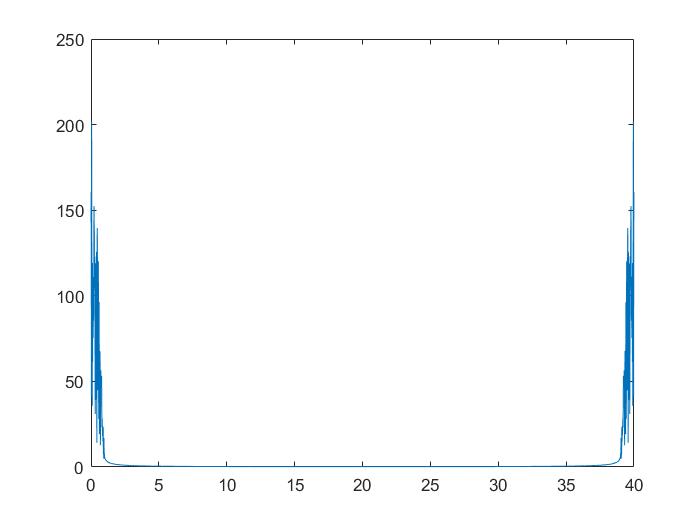


图2.5 通过低通滤波器后频谱 图2.6 调制到发送频段后频谱

接收端由PlutoSDR获取完整的20M频带中的所有信号之后发送给电脑处理，对该数据，针对设备配置的接收频段，将接收信号通过以该频段为中心频率归一化带宽为0.2的带通滤波器滤除其他频带的干扰信号。之后通过下变频将该信号变到基频后再通过一个归一化带宽为0.2的低通滤波器滤除其余的频谱分量。最后将得到的信号进行10倍抽取将码率变换到10Mbit/s来与后续的BPSK解调程序相匹配。

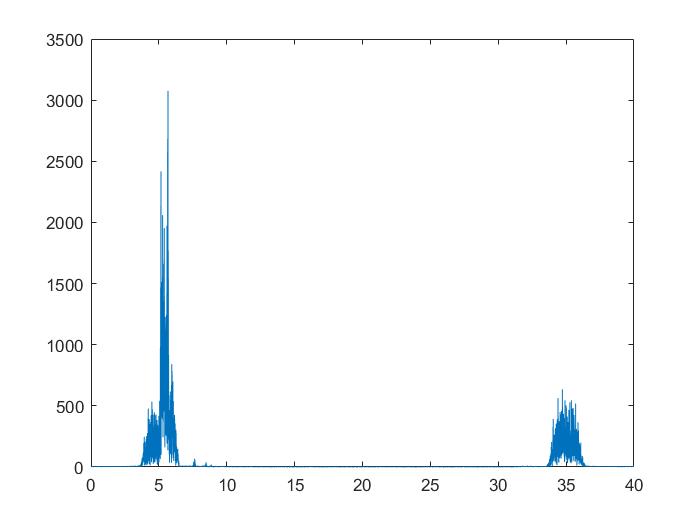
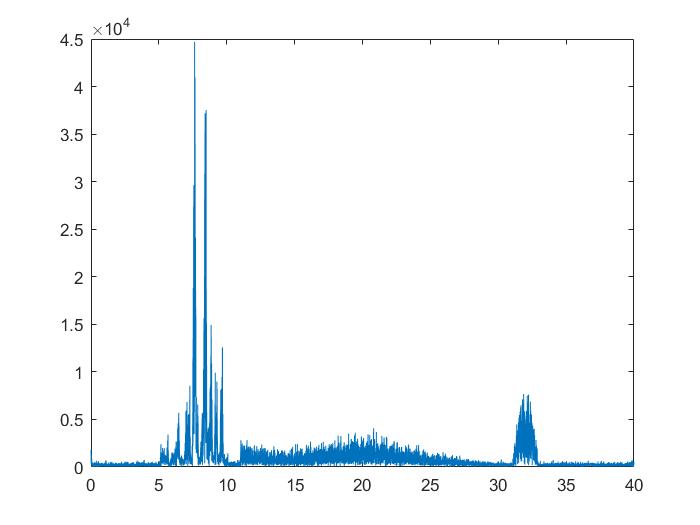


图2.7 接收信号频谱 图2.8 带通滤波后频谱

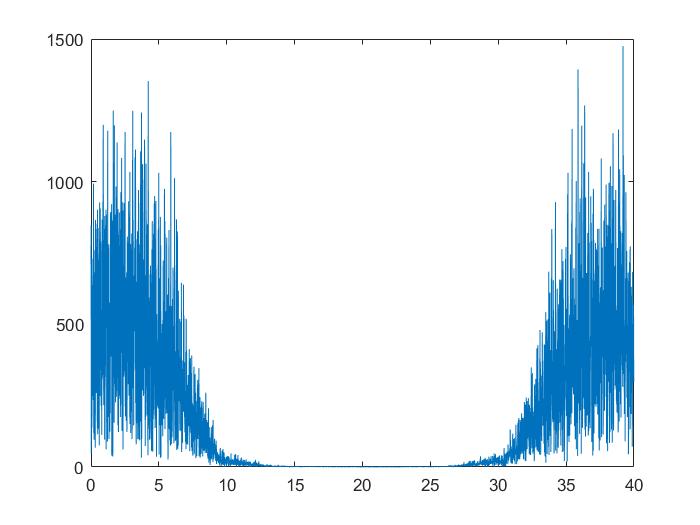
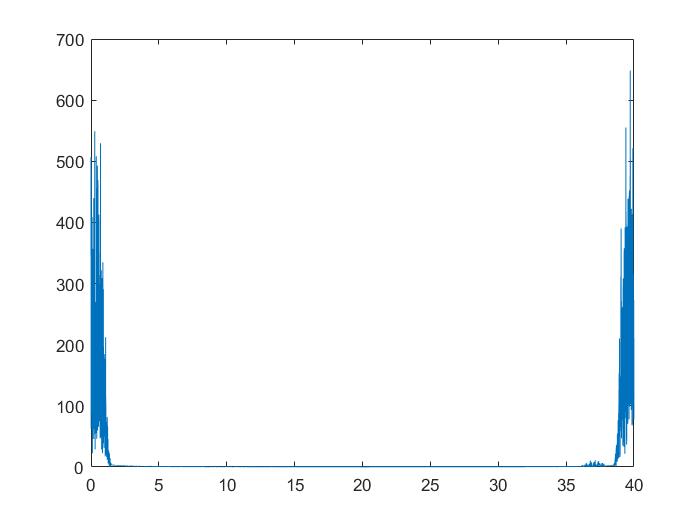


图2.9 下变频后信号频谱 图2.10 抽取后频谱

# 3．传输层设计

## 3.1 系统组网设计

## 3.1.1 数据帧格式

针对本系统设计对数据帧进行了优化，一般一个数据帧会有一个头部和一个数据包组成，而本系统每一个数据帧由两个帧头和一个数据包组成。每个帧头的长度为14字节，数据包长度可配置，默认长度为160字节。

每个帧头部的第一个字节用来指示发送目的地，第二个字节用来指示命令，共五种命令，以及12个字节的额外的头部数据。

1.发送数据包请求命令，用来发起一次传输，其头部数据的前两个个字节用来指示接下来要发送的数据包数量。接下来两个字节用来指示最后一个数据包中补零的数量。

2.发送数据包应答命令，用来回应发送数据包请求命令，通知对方我已准备号，可以开始传输数据。

3.数据包发送命令，用来进行数据包传输，其头部数据用来指示当前数据包的序号。

4.数据包确认命令，接收端用于回应发送端，自己已经接收到的数据包，由十个字节的头部数据用来指示最近五次接收到的数据包序号，以及两个字节的数据用来指示想要接收的数据包。

5.数据接收完毕命令，当接收端接收完数据之后发给发送端，用于指示当前数据已传输完毕。

## 3.1.2 组网方案

通过参数可以配置任意一台设备的发送频段和接收频段，该系统对每台设备提供了两个工作的接收频段，分别是通道1和通道2，每个通道都可以独立工作。两个通道可以配置为发送模式或者接收模式，因为发送情况下也需要一个接收通道收取回传的数据。目前支持除了双发送以外的所有工作模式。当通道工作在发送模式时，其占用发送数据帧的一个头部和数据包部分，而工作在接收模式时，只占用一个头部。

当某个设备需要发送数据到另一个设备时，将该设备的某个通道的接收频段配置为另一个设备的发送频段，并将该通道配置为发送模式。相应的，另一台设备则将其某通道的接收频段配置为此设备的发送频段，并将通道配置为接收模式。接收端会持续监听该信道，并将该信道中传输的数据帧获取下来，分析该数据帧的两个头部，若发给自己，则接收并进行处理，反之，则丢弃。通过这种方式即可实现任意数目设备之间的组网。

当两个接收频段均配置为自己的发送频段时，可以进行系统的环回测试。而当设备1的两个接收频段均配置为设备2的发送频段，而设备2的两个接收频段均配置为设备1的接收频段时，可以实现设备1和设备2的全双工通信。而当三台设备的发送频段和两个接收频段均不一样时，即可工作在三台设备的组网下。

## 3.2 端对端传输设计

在组网正确配置的情况下，由组网部分过滤后的数据即可视作端到端的直接数据包传输，可以屏蔽掉具体实现细节，可看作A生成了一个数据包，然后直接交付给了B。

但是该数据包，不一定是顺序交付的，也不一定是实时交付的，而且在交付的过程中也可能发生错误。

因此需要一个可靠的传输控制，通过该传输控制功能，能够实现最后所有A要发送的数据包均无误的发送到了B，并且B能够按照顺序将数据包进行排列。

首先，由发送端A发起一次数据传输，并告诉B将发送的数据包数量以及补零的数目。B根据要接收的数据包数量，准备一个接收指示缓存，该缓存用来指示目前已经接收到了哪些数据包，相应的，发送端也有一个发送指示缓存，用来指示B已经接收到了哪些数据包。

为了提高数据发送效率，发送端A不等待任何回传即发送下一个数据包，下一个发送的数据包是当前数据包的下一个未接收数据包，可通过查找发送指示缓存来确定下一个未接收数据包。B接收到数据包之后，根据数据包序号更新接收指示缓存，并收下数据部分，之后，将该数据包序号更新的B的接收确认命令的额外头部数据中回传给A。

A根据B的额外头部数据中指示的已接受数据包序号和待接收数据包序号更新A的发送指示缓存。当B的接收指示缓存全满之后，发送给A数据接收完毕命令，A接收该命令后停止发送，至此，一次数据传输就完成了。

# 4．应用层设计

## 4.1 wav文件传输设计

wav文件是在PC机平台上很常见的、最经典的多媒体音频文件，其采用44100HZ的采样率，16比特量化。使用Matlab的audioread函数读取wav文件得到的是double类型的浮点数，其绝对值不大于1。对其进行16bit量化后将其每个采样点的16bit数据拆分成两个8bit数据，一个用来存储其高八位，一个用来存储其低八位，拆分之后数据长度翻倍。将拆分后的数据根据数据包长拆分成多个数据包，对于最后一个数据包长度不够的部分进行补零。

接收端将接收到是数据包按照序号组装成完整的数据，然后去除后面补的零。再将两个相邻的数据重新合成一个16bit数据。最后将该16bit数据转化成double型，通过audiowrite函数写入到对应的wav文件中，即完成了wav文件的传输。

## 4.2 人机交互设计

为了方便使用者的使用，提供了良好的人机交互功能，软件流程如图4.1。

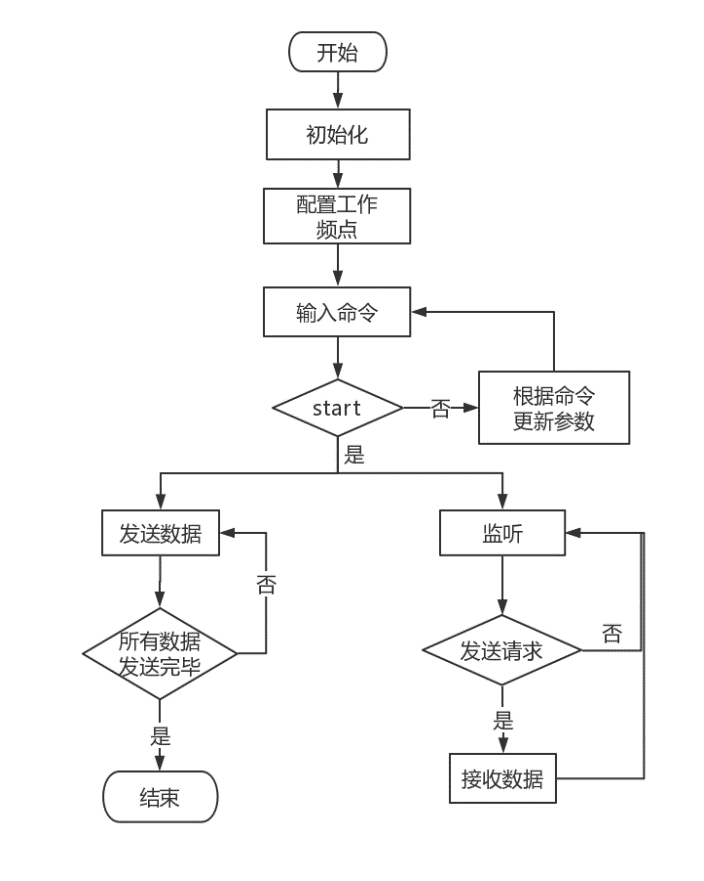


图4.1 软件流程图

该系统可通过命令配置实现通道的发送接收模式选择，也可以通过repeat命令来设置文件的多次发送。该系统还针对不同的需要，对于显示的消息设置了不同的屏蔽等级，通过设置越高的屏蔽等级来显示更少的消息。同时，也可通过配置来控制系统工作是否工作在调试模式下，调试模式下，会显示接收的数据图片和星座图，但是会占用更多的时间。

# 5. 小结

基于PlutoSDR平台搭建了一个组网通信系统，能够实现2米以上的三台设备的组网，并能够传输wav格式的音频文件，传输速率远大于64Kbit/s，误码率远小于1%，基本实现了题目的要求，不过调制方案选用的是BPSK，没有使用OFDM。

通过频分复用加上数据组装成帧，实现了通信系统的组网和端到端的可靠传输，针对该系统设计要求提供的双帧头设计能够有效提高系统传输效率，避免一台设备需要同时发射和接收时的数据争用情况。发送端的每个发送数据帧可以同时包含发送通道的数据和接收通道的数据。若使用单帧头，则发送通道和接收通道会争用同一帧数据，降低传输效率。

同时，针对在大文件数据传输中极低概率出现的接收端没有接收到某个数据包，但是发送端却认为接收到了的情况，将接收确认帧头中原来12字节用来指示接收到数据的最后两个字节改成了指示预接收的数据。发送端根据最后两个字节指示的信息清楚对应的发送指示缓存，即可消除该问题。