

应急状况下北斗短报文通信功能的应用

张志峰, 李中学

(陆军勤务学院 军事物流系, 重庆 401311)

摘要: 北斗卫星导航系统的短报文通信功能在传输容量和传输频率上具有一定的限制; 在应急情况下, 为了准确高效地进行信息的交互, 通过对短报文传输协议的改进, 提出了一套应急情况传输机制; 将信息按照紧急程度进行划分, 采取相应的处理方法, 能够在有限的传输空间下, 优先发送重要信息, 提高传输效率, 为应急状况的处理提供信息交互的保障。

关键词: 短报文通信; 传输协议; 传输机制; 数据单元

Application of Short Message Communication Function of Beidou in Emergency Situation

Zhang Zhifeng, Li Zhongxue

(Dept. of Military Logistics, Army Logistics University of PLA, Chongqing 401311, China)

Abstract: the short message communication function of the Beidou Satellite Navigation System (BDS) has a certain limit on the transmission capacity and transmission frequency. In emergency situations, in order to carry out information interaction accurately and efficiently, a set of emergency transmission mechanism is proposed by improving the short message transmission protocol. The information is divided according to the emergency degree, and the corresponding processing method is adopted. It can send priority information in the limited transmission space, increase transmission efficiency, and provide information interaction guarantee for emergency handling.

Keywords: short message communication; transmission protocol; transmission mechanism; data unit

0 引言

我国幅员辽阔, 地震、洪水、泥石流等大型自然灾害时有发生, 造成的人员伤亡和经济损失都十分巨大^[1-4]。灾害发生地区容易产生通信设备瘫痪问题, 导致链路中断、通信不畅, 加之偏远地区本身所存在的通信盲区, 无法将灾区信息及时传递出去, 对于救援工作的开展带来很大的不便^[5]。为了尽快投入救援工作, 最大程度的保护人民群众的生命财产安全, 构建应急通信链路进行信息的传递是十分必要的。

常规的通信技术由于通信原理的限制, 在应急情况下无法很好的完成通信任务^[6]。例如采用以太网进行远程数据传输, 虽然可以确保数据传输的实时性, 但基站等通信设备易遭到破坏^[7-8]; 基于手机无线自组网构建小范围应急通信系统, 虽然可以实现语音和文本的一对一或一对多通信, 但无法保证传输链路的通畅性以及数据传输的可靠性^[9-10]; 使用 ZigBee 网络和 GPRS 传输数据, 虽然可以提

高数据传输的可靠性, 但 ZigBee 网络无法进行移动终端组网, 且传输的距离较近, GPRS 存在着丢包现象和转接时延的问题^[11-14]。

相比而言, 卫星通信技术可以不受自然灾害的影响, 不存在通信盲区, 稳定性较高; 并且误码率低, 北斗系统的误码率最大为 10^{-5} , 传输可靠^[15-16]。因此, 研究应急情况下北斗短报文通信功能的应用, 构建通信系统, 实现一对一或一对多的通信, 对于灾区信息的及时发送具有现实意义。

1 信息预处理

应急情况下, 首先应当保证信息发送的时效性, 在此基础上, 考虑到北斗短报文通信功能所具有的容量和频度的限制, 为了充分利用传输空间, 尽可能多地将有用信息正确发送出去, 需要对信息进行前期处理。为了便于说明, 本文将一条信息称为一个数据单元 (Data Unit), 用符号 DU 进行表示。

处理 1: 根据数据单元所含内容重要程度的差异, 将其分为紧急数据单元和普通数据单元两类, 用符号 UDU 和 ODU 进行表示。UDU 的内容较为重要, 时效性要求比较高, 需尽快让接收方知晓; ODU 内的内容相对来说对于发送时间没有严格要求, 因此, 发送端应优先对 UDU 进行处理。

为了处理方便, 在发送端缓冲区内, 划定队列 Queue1 和队列 Queue2。如图 1 所示, Queue1 是由 UDU 组成的队

收稿日期:2018-04-12; 修回日期:2018-04-19。

基金项目:军委后勤保障部综合计划局下达的重点科研项目 (BS215R012)。

作者简介:张志峰(1993-),男,山东聊城人,在读硕士研究生,主要从事信息系统与应用方向的研究。

李中学(1965-),男,四川岳池人,博士,教授,硕士研究生导师,主要从事信息系统与应用方向的研究。

列, 具有优先处理权; Queue2 是由 ODU 组成的队列, 优先级较低。在发送信息时, 指针首先指向队列 Queue1, 当遍历 Queue1 内数据单元后, 再将指针指向队列 Queue2, 从而保证数据单元按重要程度进行发送。

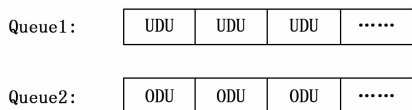


图 1 发送端缓冲区内队列划分

处理 2: 根据北斗短报文通信功能容量的限制, 将需要发送的数据单元按照所含数据量的多少进行划分。理论上北斗短报文通信功能的传输能力可达 1680bit/次, 但由于生产厂商、应用对象等因素的不同, 各北斗设备的实际传输容量存在着差异。

如图 2 所示: 假定某北斗设备的传输容量为 N bit, 设定限值区间为 $(N/2, N)$ 。DU 内所含数据量用 M 表示, 当 $M < N/2$ 时, 划分为小数据单元 (SDU); 当 $N/2 \leq M \leq N$ 时, 划分为正常数据单元 (MDU); 当 $M > N$ 时, 划分为长数据单元 (LDU)。在处理过程中, 对于 SDU 需要进行数据整合, 对于 LDU 则需要进行数据切分。

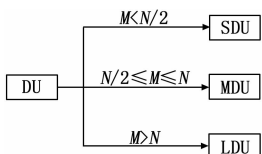


图 2 根据 DU 大小进行划分

2 通信协议设计

为了满足应急情况下应用的实际需求, 本文提出了相应的机制, 需要对北斗短报文通信协议进行一定的改进。在北斗原有协议的基础上, 添加必要的传输约定关键字, 以便于后续对数据的处理。协议改进部分的具体格式如表 1 所示。

表 1 通信协议改进部分的格式

| 起始位 | 标志位 | | | | 数据单元 (DU) | 长度位 | 终止位 |
|------|------|--------|-------|--------|-----------|------|-----|
| | 反馈标识 | 数据类型标识 | 数据序列号 | LDU 标识 | | | |
| 8bit | 1bit | 1bit | 5bit | 1bit | 8bit | 8bit | |

1) 起始位。占用一个字节, 以符号 “*” 表示。当读取到起始位标识时, 表明一条信息的开始。起始位对于 SDU 整合处理后的信息读取具有重要作用。

2) 标志位。用于传输机制的控制, 可分为四个部分: 反馈标识、数据类型标识、数据序列号、长数据单元标识。

(1) 反馈标识。占用 1 个比特, 用于表示 DU 是否为反馈数据单元。设定若反馈标识为 0, 则 DU 不是反馈信息; 若反馈标识为 1, 则 DU 为反馈信息。反馈标识用于确保传输可靠, 对发送的信息进行回复。

(2) 数据类型标识。占用 1 个比特, 用于表示 DU 所含

信息的紧急程度。设定若数据类型标识为 0, 则为 UDU; 若数据类型标识为 1, 则为 ODU。数据类型标识, 对于应急条件下优先发送紧急信息的处理具有重要作用。

(3) 数据序列号。占用 5 个比特, 对需要发送的 DU 进行编号, 并且根据划分出的队列 Queue1 和队列 Queue2 分别进行编号。这样能够使得处理过程中能够简单快捷的找到所需的数据单元, 对重传处理具有重要作用。

(4) 长数据单元标识。占用 1 个比特, 用于表示 DU 是否为 LDU。设定若长数据单元标识为 0, 则不是 LDU; 若长数据单元标识为 1, 则是 LDU。长数据单元标识对于 LDU 的切分以及接收后的重组处理具有重要作用。

3) 长度位。占用 1 个字节, 用来标识 DU 的净长度 (不包含添加的传输约束关键字)。长度位是 SDU 整合处理以及 LDU 切分处理的依据。

4) 终止位。占用一个字节, 以符号 “#” 表示。当读取到终止位标识时, 表明一条信息读取完毕。

3 发送端信息处理

3.1 UDU 的处理

对于成为 “信息孤岛” 的应急状况而言, 在有限的时间和容量条件下, 紧急情况中发送的多为 SDU 和 MDU, 对于 LDU 的使用较少, 这样可以尽可能多的将信息发送出去, 对救援工作的指挥部署和具体方案的实施提供可靠的保障。因此, 本文仅对 SDU 进行处理, 处理步骤如下:

步骤一: 顺势检验队列 Queue1 中是否还有其它 UDU。若有的话, 进一步判断其是否也为 SDU; 若没有的话, 就直接发送该 UDU;

步骤二: 若其后一个 UDU 也为 SDU, 则判断两者能否整合成一个 UDU; 若其后不是 SDU, 则直接发送第一个 UDU;

步骤三: 若两者能够整合, 则整合成一个 UDU, 进而判断整合后的 UDU 是否仍旧是 SDU; 若不能整合, 则直接发送第一个 UDU;

步骤四: 若整合后的 UDU 仍旧是 SDU, 则重复步骤一至三, 直至整合后的 UDU 不是 SDU 为止, 将整合后的 UDU 进行发送。

UDU 的处理过程如图 3 所示。

对于 UDU 而言, 为了不影响信息传输的时效性, 处理过程应当在极短的时间内完成。为防止 SDU 过多, 整合过程持续循环, 数据长时间滞留的情况发生, 在整合处理中需设定一个计时器。计时器的时间设置为北斗设备发送两次报文的时间间隔, 从而强制终止整合操作, 将数据信息发送出去。

值得注意的是, SDU 整合过程中, 对于每个 DU 添加的关键字仍然需要保留, 这样接收端在接收到数据包后, 按照协议制定的规则进行解析就能够还原出每条信息, 避免造成信息的混淆。但处理后的 DU 相比于长度位所标定的 DU 的净长度有所增加, 所以整合时应当考虑传输约定关键字的因素, 每合并一个 SDU 其真实数据量为标志位标

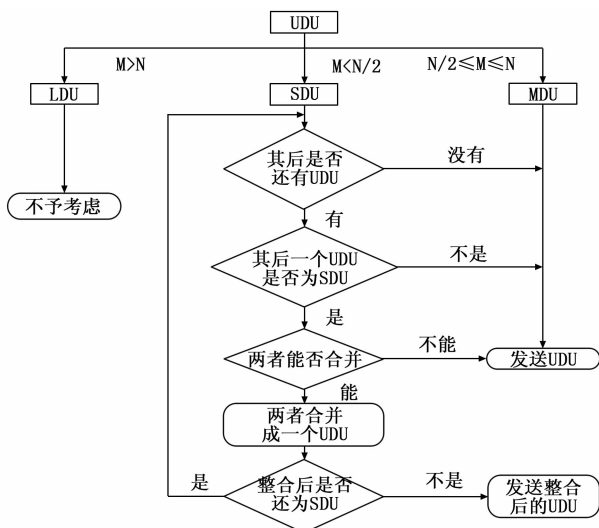


图 3 UDU 的处理

定的数据量加上关键字的长度。

3.2 ODU 的处理

相较于 UDU 仅对 SDU 进行整合处理的情况，在 ODU 中，由于时效性的要求降低，信息滞留的时间相对较长，因此数据处理的可操作性增加，不单只是 SDU 间可以进行整合，也可考虑和 MDU，甚至于 LDU 切分后的数据单元进行整合。

一些 ODU 虽然达到了 MDU 的限值，但是数据量不是特别的大，如果将其按照一个数据包进行发送，则会造成传输空间的浪费。为了尽可能充分的利用传输空间，需设定一个更加精确的限值如 $(5N/6, N)$ ，当数据量不满足限值时，虽然是 MDU，但仍需进行整合处理。

如图 4 所示，在发送端缓冲区中建立一个临时队列 Queue3，专门用来存放队列 Queue2 中的 SDU。通过这一处理过程，可以将 SDU 挑拣出来，这样在整合处理的过程中，就能有针对性的进行操作，使整合处理高效快捷。

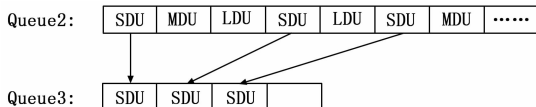


图 4 建立临时队列 Queue3

对于 ODU 的处理过程如下：

当队列 Queue2 中为 SDU 时，将其提取出来按序放入临时队列 Queue3 中。同时检查 Queue3 的长度是否达到精确限值 $(5N/6, N)$ ，若达到限值，则将 Queue3 中的 SDU 整合成一个数据包进行发送；若没有达到限值，则暂时搁置。

当队列 Queue2 中为 MDU 或者 LDU 切分后的数据单元时，步骤如下：

步骤一：让其按照 Queue3 的排列顺序从前至后进行尝试，看能否和其中的某个 SDU 进行整合；

步骤二：如果能够整合，则将两者进行合并，成为一

个 ODU，并判断整合后的 ODU 是否满足精确限值 $(5N/6, N)$ ；如果不能够整合，则直接发送该 MDU 或者 LDU 切分后的数据单元；

步骤三：若整合后的 ODU 不满足精确限值，则重复步骤一和二；若满足精确限值，则发送整合后的 ODU。

虽然 ODU 对于时效性的要求没有那么严格，但是也不应一直滞留，因此在队列 Queue3 中应该添加一个机制，当某个 SDU 滞留的时间达到一定值时，强制性进行发送。因此考虑对 SDU 添加计时器，从其进入队列 Queue3 开始计时，当滞留时间达到 30 分钟时，不再考虑数据的整合，直接进行发送。

ODU 具体的处理过程如图 5 所示。

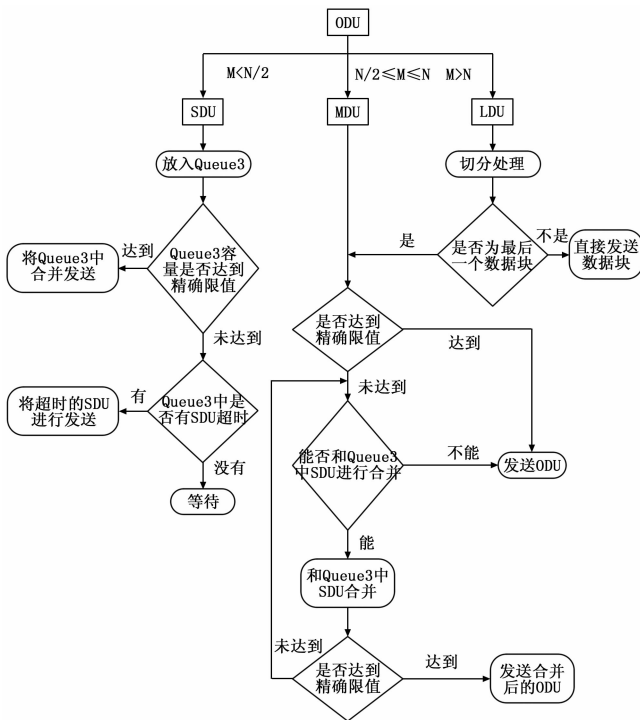


图 5 ODU 的处理

3.3 LDU 的切分处理

对于 LDU 而言，保证接收端数据的完整性是重要条件。因此，对于 LDU 的切分，要考虑切分出来数据块的大小和顺序性。为了处理简单，以 ODU 处理中设定的精确限值作为切分的依据。这样切分出来的前几个数据块都可直接发送，不必再考虑和 SDU 的整合问题。对于切分出来的最后一个数据块，应当按照 ODU 处理中提到的处理方法进行处理。

为了方便传输，需要对切分出来的数据块进行处理，使其符合制定的传输协议规则。为了便于数据块的重组，在标志位中增加一个数据块序列号，约占用 3 个比特位，其传输协议格式如表 2 所示。

LDU 切分后，第一个数据块保留有起始位和标志位，最后一个数据块保留有长度位和终止位。为了整合处理方便，现将其舍弃，重新添加关键字信息。由同一个 LDU 切

表 2 LDU 切分后的传输协议格式 bit

| 起始位 | 标志位 | | | | | 数据单元 | 长度位 | 终止位 |
|-----|------|--------|-------|--------|--------|------|-----|-----|
| | 反馈标识 | 数据类型标识 | 数据序列号 | LDU 标识 | 数据块序列号 | | | |
| 8 | 1 | 1 | 5 | 1 | 3 | | 8 | 8 |

分出来的数据块, 其起始位和标志位相同, 使用 LDU 本身的起始位和标志位信息, 仅根据实际改变数据块序列号, 这样保证了切分出来的数据块符合整个传输协议所制定的规则。在数据接收端进行 LDU 数据块的重组时, 就能够根据相同的数据序列号对其进行合并, 并通过数据块序列号进行合并顺序的确定, 确保数据重组的完整性。

4 接收端信息处理

4.1 差错检测

在北斗短报文通信的传输过程中存在着丢包的问题, 据相关研究表明, 北斗短报文通信单数据包的传输成功率约为 95.5%; 并且对于一个数据包来说存在着一定的传输出错率, 随着包内数据量的增加传输出错的可能性也会增加。因此, 接收端在收到数据包时, 首先需要进行差错检测。

针对丢包问题的检测, 通过数据包编号来进行。设定数据包编号标志位, 编号取 1~1 000, 当发送端发送数据包时, 按照从小到大的顺序对其自动进行编号。当数据包发生丢失时, 可通过其相邻的数据包编号将其查找出来。假设 46 号数据包发生丢包问题, 接收端在接收到 45 号包后, 下一个接收到的数据包编号将为 47, 这样就能够检测出 46 号包丢失。

针对传输数据出错的检测, 通过长度位关键字来进行。接收端收到数据包后, 对其进行解析, 通过计算 DU 的长度和长度位进行比较, 如果结果一致, 则表明 DU 传输正确, 若不一致, 则检测出传输错误。

这种方法仅为一种简单快速的检验, 并不能精确的将所有错误检测出来, 但是相比于其他差错检测方法而言, 具有所需添加的冗余信息少, 检测时间短的优势, 在应急情况下具有一定的应用价值。

4.2 反馈机制

北斗短报文通信从本质上来说属于不可靠通信, 再加上丢包和传输数据出错等因素, 发送端不能明确的得知报文是否发送成功。在应急情况下, 信息的交互显得至关重要, 为了让通信双方能够知晓信息发送结果, 充分利用其双向通信的性质, 采用最原始的逐条反馈机制, 对接收到的每一个数据包都发送一个反馈数据包。

接收端接收到数据包后, 首先对其进行差错检测, 然后根据检测结果, 向发送端发送反馈信息, 告知发送端数据包是否发送成功, 数据传输是否完整, 以便发送端采取相应的措施。反馈信息作为一个 UDU 存在, 在接收端按照 UDU 发送方法进行操作。

当发生数据包丢失时, 反馈信息中只需说明数据包编

号即可; 当发生数据单元传输出错时, 需要发送数据类型标识以及数据序列号, 并且若为 LDU 出错, 还需要发送长数据单元标识以及数据块序列号。这样发送端在接收到反馈信息时, 就能确定发送出错的原因, 并定位到出错位置。

4.3 数据重传处理

4.3.1 丢包问题的处理

通信系统中解决数据丢包的问题, 主要采用自动重传请求技术和前向纠错技术两种^[17]。自动重传请求技术根据反馈数据包的内容, 对丢失的数据包进行重新发送; 前向纠错技术通过增加冗余数据来实现数据的恢复。

前向纠错技术虽然具有时延低和恢复丢包效率高的优点, 但是其操作过程更加复杂, 并且需要加入冗余数据包, 这对于传输信道极其珍贵的应急情况而言, 并不适用。相比较而言, 重传技术简单而且不会带来过多的额外信息传输, 因此本文采用重传技术处理丢包问题。

当通过反馈信息得知数据包丢失时, 接收端根据数据类型决定是否进行重发。对于紧急数据类型的数据包, 信息重要程度较高, 应立即进行重发; 对于普通数据类型的数据包, 则可以根据信息的内容有选择的进行重发, 那些可有可无的信息将不再进行重发, 以节约传输信道空间。

4.3.2 传输数据出错处理

发送端在读取到反馈信息时, 根据出现错误的数据单元类型采取相应的处理。当 UDU 出现错误时, 将其进行按照紧急数据类型的处理方式重新发送; 当 ODU 出现错误时, 由发送者决定是否进行重传。

这种处理方法对于 SDU 出错的情况有很好的效果, 不必将由该 SDU 整合而成的数据包进行全部的重传, 节约了传输信道空间, 同时避免了重复性传输信息, 在应急情况下具有较高的实用性。

5 实验结果与分析

本文采用 Matlab 软件对所提出的传输机制进行模拟仿真, 主要对数据类型分类以及 DU 的处理方法进行了验证。如图 6 所示, 引入元胞数组的概念, 将每一行看作是一个 DU, 某一时刻发送端所有需要发送的 DU 看作是一个元胞数组。第一列模拟 DU 的大小, 第二列用“急”和“普”来标识 UDU 和 ODU, 第三列用数字来表示数据单元形成的先后顺序。为提高仿真效率, 本实验中设置 $N=30$ 。

为了模拟现实情况中所需处理的 DU 的任意性, 在形成第一列的元胞数据时, 首先通过 ' $A = \text{randint}(1, i)$ ' 产生由 0 和 1 两个数组成的 $1 * i$ 维矩阵; 然后通过循环产生 30 个长度不固定的 $1 * i$ 维矩阵, 分别组成 UDU 和 ODU 两个元胞数组, 并在第二列进行标识; 最后通过 ' $r = 1 + 30 * \text{rand}([1 10])$ ', 产生 10 个 (1, 30) 范围内均匀分布的随机数, 通过 ' $r1 = \text{fix}(r)$ ' 对随机数取整, 并按照产生的随机数从两个元胞数组中分别抽取相应的 10 行组成图 6 所示元胞数组, 在第三列进行编号。运用对 UDU 和 ODU 的处理方法, 对元胞数组进行编程处理, 得到的处理结果

(下转第 285 页)