

北证办发〔2025〕75号附件4

北京证券交易所、
全国中小企业股份转让系统
组播行情数据分发协议
(V1.0)

北京证券交易所
全国中小企业股份转让系统
二〇二五年七月

名词释义

词汇缩写	含义
BJQG	Beijing Market Quotation GateWay, 北京证券交易所及全国中小企业股份转让系统行情网关
MDDP	Market Data Distribution Protocol, 组播行情数据分发协议, 行情网关接入方式配置为组播时使用的分发协议
VSS	Vendor Supplied System, 供应商提供的行情接收系统, 用于接入行情网关接收行情数据

目 录

一、 前言	1
二、 类型定义	1
三、 报文结构	2
3.1. 报文头	2
3.2. 报文体	6
3.3. 报文尾	7
四、 管理报文定义	7
4.1. 组播心跳报文	7
4.2. 数据流心跳报文	8
4.3. 数据流结束报文	9
五、 附录	10
5.1. 发送者-组播流-数据流-数据包关系图	10
5.2. 应用消息解码	10
5.3. SenderId 编码机制	11
5.4. SeqNum 检测机制	12
5.5. 数据源故障检测	13
5.6. 公告	13

北京证券交易所、全国中小企业股份转让系统 组播行情数据分发协议

一、前言

组播行情数据分发协议（以下简称 MDDP）是一套适用于北京证券交易所及全国中小企业股份转让系统行情数据分发的传输控制协议。MDDP 的主要目的在于以更加高效和便捷的方式为市场参与人的多套业务系统提供实时行情分发服务。

本协议主要关注如何使用 MDDP 进行数据传输，应用层消息的相关定义参见《北京证券交易所、全国中小企业股份转让系统行情数据接口规范（二进制）》。

二、类型定义

MDDP 使用的所有整数类型均采用网络字节序（BIG-ENDIAN）进行传输。

表 2-1 类型定义

类型	描述
Int8	8位有符号整数
Int16	16位有符号整数
Int32	32位有符号整数
Int64	64位有符号整数
UInt8	8位无符号整数
UInt16	16位无符号整数
UInt32	32位无符号整数
UInt64	64位无符号整数
Char[x]	Char[x]表示字符串，x表示字符串最大字节数，字符串实际长度小于字段类型最大长度的右补空格；字符串统一使用GB18030编码。

VarChar	变长字符串，长度由前导的VarCharLength指定；字符串统一使用GB18030编码。
VarCharLength	变长字符串长度，uInt32
RawData	变长二进制数据，长度由前导的RawDataLength指定
RawDataLength	变长二进制数据长度，uInt32
Padding[x]	无格式数据，填充位

三、报文结构

本章主要描述 MDDP 协议的组播报文定义。每个 MDDP 组播报文都由报文头、报文体、报文尾三部分构成：

表 3-1 报文结构

报文头	报文体	报文尾
-----	-----	-----

3.1. 报文头

报文头不加密传输，格式如下：

表 3-2 报文头

域名	类型	长度 (byte)	意义
Protocol	uInt8	1	协议标识，固定为0xFF
Version	uInt8	1	版本号，当前为0x01
HeaderSize	uInt8	1	整个报文头的长度，指示报文头由几个4Byte words组成。
SenderId	uInt8	1	发送源标识
MarketId	uInt16	2	市场标识
Channel	uInt16	2	频道号，同一市场内的频道不

北京证券交易所 全国中小企业股份转让系统
组播行情数据分发协议

			重复，同一频道内的数据内容有序。
SeqNum	Int64	8	报文序列号
MsgCount	uInt16	2	报文内消息条目数
Flag	uInt16	2	报文标志位
OriginalSize	uInt32	4	原始数据大小（可选）
CompressedSize	uInt32	4	压缩后数据大小（可选）
EncryptedSize	uInt32	4	加密后数据大小（可选）
Padding	Byte[x]	x	字节对齐填充位（可选），确保报文头长度为4字节整数倍，无意义。

1. HeaderSize

整个报文头的长度,长度计算从报文头的第一个字段 Protocol 开始到最后一个字段 Padding 结束(如 HeaderSize=5 表示报文头总长度为 5*4Byte=20Bytes)。

2. SenderId

发送源标识，用于识别同一个组播地址及端口上的不同发送者。

3. MarketId

市场标识，预留字段，目前固定填 1。

4. Channel

行情频道，行情数据可根据业务（如按证券集）分为不同的类别，每个类别的行情数据则可根据其数据量大小进一步分为多个频道。

5. SeqNum

报文序号，每个发送者会在对应市场的 Channel 组合内维持一个从 1 开始递增的序列号，报文内的每一条消息都会让序列号递增 1，下一个报文的 SeqNum 为上

一个报文最后一笔消息的 SeqNum+1。根据 ResendBySeqNum 标志位的不同, SeqNum 有两种编码机制:

ResendBySeqNum=0, 发送者启动或重启后的首个报文从 1 开始编号。

ResendBySeqNum=1, 发送者使用报文内第一条消息的序列号作为报文序号。

(应用消息存在序号并连续递增时才可以使用该模式, 如逐笔行情)

6. MsgCount

该字段指示报文体中消息条目数。

7. Flag

Flag 标志位, 用于标识报文属性, 左起第一位为 Bit15。

表 3-3 Flag 标志位

Flag Bit	意义
Bit 15	PossDupFlag标志位, 可能重复标志 标识报文中的数据为可能重复的数据, 此时 SeqNum可以小于期望的序列号。数据接收者如果已经接收到该数据则可以直接丢弃。 0: 非可能重复报文 1: 可能重复的报文
Bit 14-13	PacketType标志位, 报文类型标志 00: 管理报文 01: 应用报文
Bit 12	ResendBySeqNum标志位 是否支持按报文的SeqNum进行重传 0:不支持 1:支持
Bit 11-10	Compress标志位, 压缩标志

	00:不压缩 01:zlib压缩 10:预留 11:预留
Bit 9-8	Encryption标志位, 加密标志 00:不加密 01:自定义加密 10:预留 11:预留
Bit 7	MsgHeader标志位,前导长度标志位 0: 前导长度不存在 1: 前导长度存在
Bit 6-0	预留

8. OriginalSize

可选字段。报文体压缩加密前的长度，报文体有做压缩或加密时，报文头里包含这个字段。

9. CompressedSize

可选字段。传输数据被压缩后的数据长度，数据有压缩时，包头里包含这个字段。

10. EncryptedSize

可选字段。传输数据加密后的数据长度，数据有加密时，包头包含这个字段。

11. Padding

可选字段。报文头长度不为 4Byte 的整数倍时，包头包含这个字段。

3.2. 报文体

报文体由 MDDP 消息头和 MDDP 消息体两部分构成，MDDP 消息头指示了报文中每一条应用消息的长度，MDDP 消息体存储了具体的应用消息。一个 MDDP 报文可以包含多条完整的应用消息，消息不跨报文传输。由于有些协议采用了上下文相关的编码机制，无法进行消息切割，因此 MDDP 消息头为可选项，由报文头中的 MsgHeader 标志位标识。

MsgHeader=1 的 MDDP 报文体格式如下：

表 3-4 MsgHeader=1 的 MDDP 报文体格式

域名	类型	长度 (byte)	意义
Lengths	uInt32[x]	4*MsgCount	MDDP消息头，指示了报文中每一条消息的长度
Body	Byte[x]	x由Lengths指定	MDDP消息体

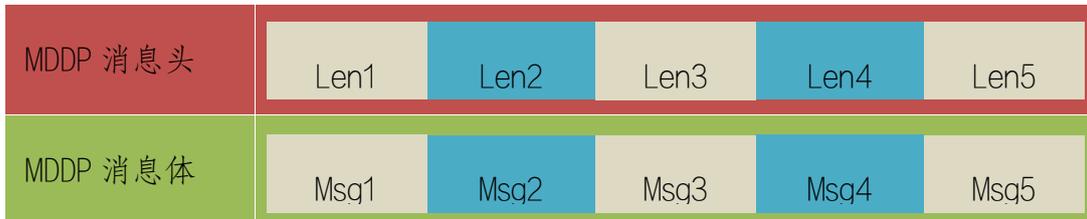


图 3-1 MsgHeader=1 的 MDDP 报文体格式

MsgHeader=0 的 MDDP 报文体格式如下：



图 3-2 MsgHeader=0 的 MDDP 报文体格式

3.3. 报文尾

报文尾定义了校验和，不加密传输。校验和的计算范围从报文头开始（包括报文头）一直到报文体结束。校验和算法为 Adler32。

表 3-5 报文尾

域名	类型	长度 (byte)	意义
Checksum	uint32	4	校验和

四、管理报文定义

本章主要描述 MDDP 协议使用的管理报文定义。

4.1. 组播心跳报文

组播流由组播地址、端口及 SenderId 唯一标识，当组播流上没有任何应用层数据时，发送者会在组播流上周期性的发送组播心跳报文，帮助数据接收者检测组播通道状态。组播心跳报文的发送间隔为 5 秒，数据接收者在 3 个心跳周期内没有收到任何数据时，说明该路组播源故障。

Channel 为 0 时表示该报文为“组播心跳报文”，组播心跳报文没有报文体，收到后可以直接丢弃。报文样例：

表 4-1 组播心跳报文样例

域名	类型	取值
Protocol	uint8	协议标识，固定为 0xFF
Version	uint8	版本号，当前为 0x01
HeaderSize	uint8	5
SenderId	uint8	发送源标识
MarketId	uint16	1
Channel	uint16	固定为 0
SeqNum	int64	0

MsgCount	uInt16	0
Flag	uInt16	高位 → 低位 0000000000000000
CheckSum	uInt32	校验和

4.2. 数据流心跳报文

数据流由 Channel 及 SenderId 唯一标识，当报文的 ResendBySeqNum 标志位为 1 时，发送者会在数据流上周期性的数据流心跳报文，用于告知数据接收者当前数据流上已发送的最后一笔数据消息对应的序号。

本协议将“数据流心跳报文”的 SeqNum 设置为当前数据流上已发送的最后一笔消息的 SeqNum。当数据流上没有发送过任何应用消息时，“数据流心跳报文”的 SeqNum 为 0。

Channel 不为 0 且 MsgCount 为 0 时表示该报文为“数据流心跳报文”，数据流心跳报文没有报文体。报文样例：

表 4-2 逐笔行情的逐笔频道心跳消息样例

域名	类型	取值
Protocol	uInt8	协议标识，固定为0xFF
Version	uInt8	版本号，当前为0x01
HeaderSize	uInt8	5
SenderId	uInt8	发送源标识
MarketId	uInt16	1
Channel	uInt16	2011
SeqNum	Int64	9999
MsgCount	uInt16	0
Flag	uInt16	高位 → 低位

		000000000000000000
Checksum	uint32	校验和

4.3. 数据流结束报文

“数据流结束报文”为特殊的数据流心跳报文，用于告知数据接收者当前数据流上所有业务数据已经发送完毕。

Channel 不为 0 且 MsgCount 为 0xFFFF (或写作 65535) 时表示该报文为“数据流结束报文”。消息样例：

表 4-3 数据流结束报文样例

域名	类型	取值
Protocol	uint8	协议标识，固定为0xFF
Version	uint8	版本号，当前为0x01
HeaderSize	uint8	5
SenderId	uint8	发送源标识
MarketId	uint16	1
Channel	uint16	2011
SeqNum	int64	9999
MsgCount	uint16	0xFFFF
Flag	uint16	高位 → 低位 0000000000000000
Checksum	uint32	校验和

五、附录

5.1. 发送者-组播流-数据流-数据包关系图

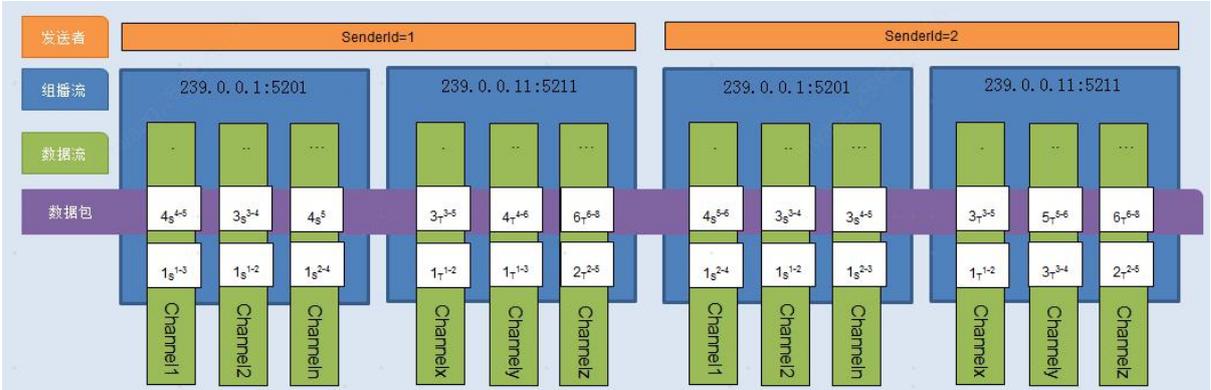


图 5-1 发送者-组播流-数据流-数据包关系图

注 1:Sender1, Sender2 为互为备份的两个发送者

注 2:组播流以组播地址、端口及 Sender 为单位

注 3:数据流以组播流及频道为单位

注 4: N_s^{x-y} , N 表示包序号, S 表示快照数据流, x-y 本数据流上的第 x 到第 y 笔快照消息, ResendBySeqNum 标志位为 0

注 5: N_T^{x-y} , N 表示包序号, T 表示逐笔数据流, x-y 本数据流上的第 x 到第 y 笔逐笔消息, ResendBySeqNum 标志位为 1

5.2. 应用消息解码

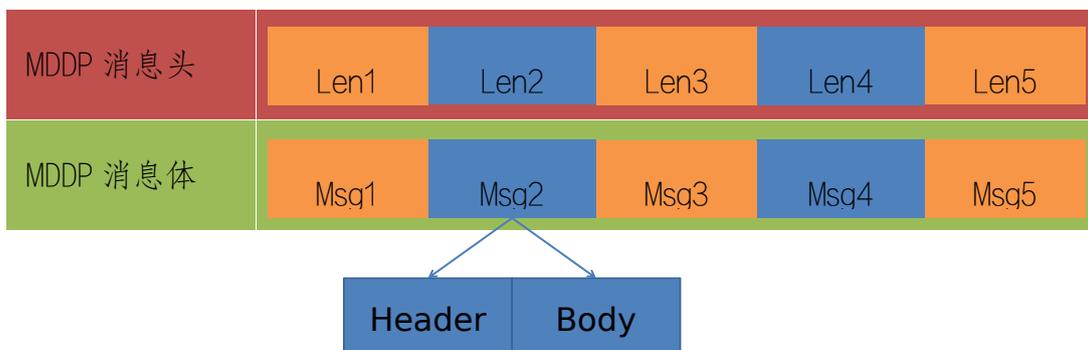


图 5-2 MDDP 消息结构

MDDP 协议当前仅支持分发是二进制行情数据，MDDP 消息体内存储的应用消息定义参见《北京证券交易所、全国中小企业股份转让系统行情数据接口规范（二进制）》。

MDDP 消息头提供了 MDDP 消息体内每一条消息的长度，接收者可以根据 MDDP 消息头提供的消息长度信息，在不解析 MDDP 消息体的情况下，实现消息的切割。

表 5-1 消息类型偏移量

消息	消息偏移量
Msg1	Offset1=0
Msg2	Offset2= Offset1+Len1
Msg3	Offset3= Offset2+Len2
Msg4	Offset4= Offset3+Len3
Msg5	Offset5= Offset4+Len4

为了提升效率，MDDP 取消了《北京证券交易所、全国中小企业股份转让系统行情数据接口规范（二进制）》中每条消息尾部的校验和，通过 MDDP 报文头的校验和对整个报文进行完整性校验。接收者对 MDDP 报文进行完整性校验后，直接通过 MDDP 消息头完成消息切割并解码消息即可，无需对每一条消息再进行单独的完整性校验。

5.3. SenderId 编码机制

SenderId 编码机制如下：

1. 交易日首次启动时，SenderId 设置为 Sender 在集群中的索引
2. 每次重启，SenderId= Sender 在集群中的索引+(启动次数-1)* 集群的 Sender 数量
3. 当 SenderId> 255 时，SenderId 重置为 Sender 在集群中的索引

SenderId 编码机制示例如下：

表 5-2 SenderId 编码机制示例

集群	发送者索引	第1次启动	第2次启动	第N次启动
Sender1	0	0	2	$0+(N-1)*2$
Sender2	1	1	3	$1+(N-1)*2$

注:如果交易日内重启次数过多或硬件故障导致重启后 SenderId 无法从磁盘恢复, 此时 SenderId 重置为 Sender 在集群中的索引。

5.4. SeqNum 检测机制

数据流上每个组播报文的序号连续递增, 由于组播会发生乱序、丢包和重包, 因此数据接收者应该对数据流上的组播报文序号进行检查。

1. $SeqNum < \text{NextExpectedSeqNum}$, 此时认为数据报文为过期数据直接丢弃。
2. $SeqNum = \text{NextExpectedSeqNum}$, 则处理报文, 并设置下一个报文期望的报文序号 $\text{NextExpectedSeqNum} = SeqNum + \text{MsgCount}$ 。
3. $SeqNum > \text{NextExpectedSeqNum}$, 说明可能发生了乱序或丢包, 可采用两种处理机制:

- (1) 直接判定为网络丢包, 从当前数据包开始继续处理后续报文。网络基础设施稳定, 很少出现乱序时, 可以考虑使用该机制。
- (2) 先判定网络发生乱序并缓存数据, 当缓存的数据超过预设的缓冲区大小或超时后依然未收到预期的报文, 则判定为网络丢包。使用缓存中序号最小报文的作为 $\text{NextExpectedSeqNum}$, 继续处理缓存中以及后续的报文。

注 1: 接收者的乱序缓冲区需根据网络乱序情况进行设置, 如果网络基础设施稳定, 很少出现乱序时, 则可以设置为一个比较小的值 (如 16)

注 2: 可重传的数据发生丢包时, 可使用 BJQG 提供的应用层重传机制进行恢复

5.5. 数据源故障检测

数据源故障重启后,快照类组播报文(ResendBySeqNum 标志位为 0)的 SeqNum 会从 1 开始重新编号,从而导致组播报文的 SeqNum 回退。MDDP 提供两种故障检测机制,便于接收者对数据源故障进行识别并进行恢复。

1. SenderId 发生变更

当接收者发现数据流上的 SenderId 发生变更时,说明数据源发生重启或发生了交易日切换,此时应使用最新的数据报文中的消息序号重置下一个期望的组播报文序号。

2. SenderId 未发生变更,但 SeqNum 回退

该场景一般是因为硬件故障导致 SenderId 无法恢复,从而导致重启后 SenderId 未发生变化但 SeqNum 发生了回退。由于组播可能发生乱序,因此接收者可配置一个阈值,当 $\text{SeqNum} + \text{阈值} < \text{NextExpectedSeqNum}$ 则认为是数据源重启,此时应使用最新的数据报文中的消息序号中重置下一个期望的组播报文序号。否则,则将报文认为是乱序包进行处理。

5.6. 公告

MDDP 协议支持公告落地功能,可通过 BJQG 提供的公告落地开关启动或关闭。公告落地功能启用后,MDDP 将公告分为公告摘要和公告内容两部分发送,其中公告摘要通过组播报文发送,公告摘要消息定义与二进制数据接口规范中的公告消息一致,公告摘要中不携带公告内容,因此 RawDataLength 被固定置为 0。公告内容则通过文件的方式存储在接收者指定的目录。



图 5-3 公告消息流程图