|  |
| --- |
| [键入公司名称] |
| 上期全挡行情 |
| my\_quote\_shfe\_my\_jr\_ex-spd\_lib |

|  |
| --- |
| wangying  2017/4/24 |

## 丢包处理

## 丢包场景

* 一个行情批次

1. 丢掉了一个行情批次中的买方向前面一些数据包
2. 丢掉一个行情批次中一些买方向数据中的中间数据包
3. 丢掉一个行情批次中买方向数据的后面一些数据包
4. 丢掉一个行情批次中中间一些买卖数据包
5. 丢掉一个行情批次中卖方向的开头一些数据包
6. 丢掉一个行情批次中卖方向数据中间一些数据包
7. 丢掉一个行情批次中卖方向数据后边一些数据包
8. 一个行情批次中所有卖方向数据丢包
9. 一个行情批次中部分买和所有卖方向数据丢包
10. 一个行情批次中所有买方向和部分卖数据丢包
11. 一个行情批次中所有买方向数据包丢失

* 跨行情批次

1. 跨批次丢失了中间部分的卖方向和部分买方向数据
2. 跨批次丢失了中间全部的卖方向和部分买方向数据
3. 跨批次丢失了中间部分的卖方向和全部买方向数据
4. 跨批次丢失了中间全部的卖方向和全部买方向数据
5. 跨批次丢失了中间部分的买卖方向和买方向数据
6. 跨批次丢失了中间部分的买卖方向和全部买方向数据
7. 跨批次丢失了中间部分的买卖方向和买卖方向数据
8. 跨批次丢失了中间部分的卖方向和买卖方向数据
9. A1709[120] 则后边还有数据
10. 前边有a1709[120] 但是丢包，后边收到a1709[30]，则无法判断丢包

如何确保数据的完整性：

前置条件：在一个批次完整数据队列中，买方向队列的所有合约在卖方向队列中也都存在（一字涨/跌停，可能因为没有买方数据，或者没有卖方向数据而被丢掉）

确保数据完整的要点：

* + - 开始接收时的起点必须是完整数据
    - 买方向缓存队列中的数据必须是完整的数据
    - 只有接收一个合约的完整数据才将该合约的买方向和卖方向的数据按顺序发出
    - 只有买方向和卖方向都有数据才认为是完整数据（这样会误丢涨停和跌停数据，但可以通过一档数据获取涨/跌停信息）

1. 找到数据接收的起点

因为系统可能是中途启动，或则已启动就有丢包发生，所以开始接收的数据可能是不完整的，为了找到系统接收完整数据的起点，我们需要执行如下步骤：

1. 接收到第一笔买方向的数据，将该数据记录为买方向victim

系统启动后，直到接收第一个买方向数据（之前的数据都丢掉），并将该合约数据标记为无效数据，之后收到另一个合约的数据，才正式开始，且这期间没有丢包发生

1. 接收买方向数据，并保存在队列中
2. 在接收到买方向数据后，先判断是否有丢包，如果有，则丢掉丢包前后边界的一个合约的数据，确保放到队列中的数据是完整得到，分丢掉前边一个合约和丢掉后边一个合约2各地方：
   * 丢掉前边一个合约

在卖方向数据队列中，将尾部最后的那个合约的所有数据设置成无效数据

* + 丢掉后边一个合约

发现有丢包后，记录丢包后的第一个合约，标记该合约数据都为无效数据，直到收到另外一个合约

1. 在收到卖方向数据后，如果有丢包，则将该合约数据标记为无效数据，直到收到另外一个合约的数据

如上，会保证买方向队列中存放的是完整数据（不完整的数据都丢掉了）；收到了完整的卖方向数据（所谓完整数据，是指接收完同一合约的所有数据，因一个包只能存120条数据）后，则将对应合约的买方向数据按顺序发送后，再按顺序发送卖方向数据

数据处理流程：

买方向数据：放到队列中

卖方向数据：count==120，则放到队列，直到收到该合约的count <120的数据，或另外合约的数据，则将当前合约的所有买队列的数据按顺序发送，之后把该合约的所有卖方向数据全发出