HKEX
香港交易所

INTERFACE SPECIFICATIONS

HKEX Orion Market Data Platform
China Connect (Securities)

Version: 1.4
22 Apr 2022

DOCUMENT HISTORY

Distribution Version

<table>
<tr><th>Version</th><th>Date of Issue</th><th>Comments</th></tr>
<tr><td>1.0</td><td>13 Dec 2016</td><td>First Distribution Version</td></tr>
<tr><td>1.1</td><td>19 Jan 2017</td><td>Revised Edition with the following updates
- Revise Short Name for Shanghai Stock Exchange</td></tr>
<tr><td>1.2</td><td>22 Mar 2018</td><td>Revised Edition with the following updates
- Section 2.2.1 – Revise description to clarify Start of Day
- Section 3.9.1 – Clarify description for BidPrice, AskPrice, AggregateBidQuantity and AggregateAskQuantity fields of message Top Of Book (655) in call auction period
- Section 3.8.1 – Revise description for dissemination condition</td></tr>
<tr><td>1.3</td><td>27 Sep 2018</td><td>Revised Edition with the following updates
- Section 3.7.2 – Switch DelistingDate field as Filler and Remove BOND and TRST as possible value for InstrumentType field</td></tr>
<tr><td>1.4</td><td>22 Apr 2022</td><td>Revised Edition with the following updates
<table>
<tr><th>Effective Date</th><th>Changes</th></tr>
<tr><td>Immediate</td><td><b>Housekeeping Update</b>
Section 2.2.3 – Update the OMD-CC shut down time as 5:00pm</td></tr>
<tr><td>Upon the launch of the inclusion of ETFs in Stock Connect</td><td><b>Update on Inclusion of exchange-traded funds (ETFs) in Stock Connect</b>
• Section 3.7.2 – Added TRST as possible value for InstrumentType field
• Section 3.8.1 – Support TradingPhaseCode for securities of ShenZhen Stock Exchange</td></tr>
</table>
</td></tr>
</table>

CONTENTS

1. INTRODUCTION

1.1 PURPOSE

This document specifies the Binary interface of the HKEX Orion Market Data Platform – China Connect (Securities) Datafeed Products (“OMD-CC”).

This document is the Transmission Specification(s) of the relevant Datafeed(s) under the standard terms and conditions (“T&C”) for China Connect Exchange Participant (“CCEPs”) and the agreement (“Agreement”) for Application Service Providers (“ASPs”) in relation to the use of China Connect market data. Please refer to Section 1.2, the summary table at Section 1.3 and Section 1.4 for the information applicable to the Datafeed(s) under the T&C and Agreement.

HKEX endeavors to ensure the accuracy and reliability of the information provided in this interface specification, but takes no responsibility for any errors or omissions or for any losses arising from decisions, action, or inaction based on this information. The CCEPs and ASPs shall not use such interface specifications for any purpose other than as expressly permitted under the T&C and Agreement. No part of this document may be copied, distributed, transmitted, transcribed, stored in a retrieval system, translated into any human or computer language, or disclosed to third parties without written permission from HKEX-IS.

1.2 READING GUIDE

The chapters following this introduction are:

Chapter 2: System Overview

Chapter 3: Message Formats

Chapter 4: Recovery

All chapters and appendices except Chapter 3 are applicable to all Datafeeds unless otherwise specified. In Chapter 3, there are indications* in individual sections/sub-sections for their applicability to individual Datafeeds. The information is also summarised in Section 1.3 Summary Table.

* Example

Section	OMD-CC SSE	OMD-CC SZSE
3.3	●	●

1.3 SUMMARY TABLE

Section	Message Format	OMD-CC SSE	OMD-CC SZSE
3.1	Data Types	●	●
3.2	Packet Structure	●	●
3.3	Packet Header	●	●
3.4	Control Messages	●	●
3.5	Retransmission	●	●
3.6	Refresh	●	●
3.7	Reference Data	●	●
3.8	Status Data	●	●
3.9	Price Data	●	●
3.10	Value Added Data	●	●

● The information supplied in the corresponding sub-section applies to the Datafeed(s)

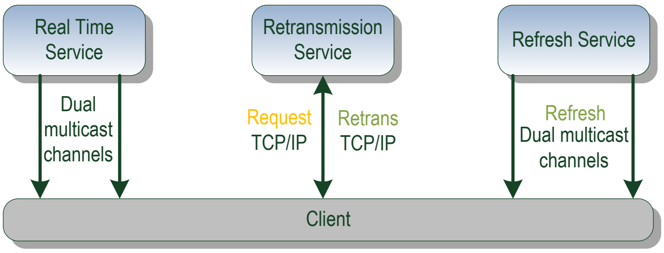
1.4 SCOPE OF INFORMATION

OMD-CC disseminates updates on market data of instruments eligible for Northbound trading under Shanghai-Hong Kong Stock Connect and Shenzhen-Hong Kong Stock Connect at intervals as provided by Shanghai Stock Exchange (“SSE”) and Shenzhen Stock Exchange (“SZSE”) respectively to CCEPs and ASPs.

2. SYSTEM OVERVIEW

2.1 SCOPE

Figure 1: Access to Market Data



OMD-CC provides market data represented in an efficient binary message format. It has been designed for high throughput and low latency. Market data of SSE and SZSE are segregated and provided in their own dedicated channels.

2.1.1 Multicast

Messages are published in a one-to-many fashion using the IP multicast and UDP transport protocols. Multicast is not a connection-oriented protocol. Data is sent strictly in one direction from server to clients.

2.1.2 Dual Multicast Channels

Due to the inherently unreliable nature of the UDP transport, packets may be lost or delivered out-of-order. To mitigate the risk of packet loss, the messages are duplicated and sent over two separate multicast channels (dual channels). Technically, a multicast channel corresponds to a multicast group.

Each pair of dual multicast channels has a unique identifier, which is referred to as the ChannelID.

More details regarding the configuration parameters (including IP addresses, port numbers corresponding to the multicast channels) are provided in the OMD-CC Connectivity Guide.

2.1.3 Recovery Mechanisms

OMD-CC provides two recovery mechanisms:

- A retransmission server provides on request gap-fill retransmission of lost messages. The retransmission requests and gap-fill replies are point-to-point (TCP/IP connection).

- A refresh server provides snapshots of the market state at regular intervals throughout the business day. Snapshots are sent using multicast on separate channels for the real time messages.

2.2 SESSION MANAGEMENT

Each multicast channel maintains its own session. A session is limited to one business day. During this day the message sequence number is strictly increasing and therefore unique within the channel.

OMD-CC does not operate on non-trading days of Hong Kong. HKEX may perform system testing on Saturdays, Sundays or days when OMD-CC is not in operation. Clients should treat data transmitted via OMD-CC on those days as non-production data and disregard them.

2.2.1 Start of Day

Housekeeping and system maintenance work may take place overnight until 6:00am. In this regard, Clients are advised to make connection to OMD-CC at or after 6:00am every business day to ensure that the data received from OMD-CC are good for the start of the day. Please also refer to the OMD-CC Developer's Guide for more information.

On each channel the first message at the start of the business day is the Sequence Reset message. The Sequence Reset message carries sequence number 1. On receipt of this message, the client must clear all cached data for all instruments.

The reference data is published each day shortly after the start of day.

If a client starts listening after the start of business day and misses the Sequence Reset message and reference data, it must use the refresh service to recover and synchronize with the real time channels.

2.2.2 Normal Transmission

Normal message transmission is expected between when the market opens for trading and when the market is closed. Heartbeats are sent around every 2 seconds on each channel when there is no activity.

2.2.3 End of Day

OMD-CC will normally be shut down at 5:00pm. This shutdown time, however, is not rigid and the Exchange has the right to adjust this time according to the different trading situations.

At the end of the business day, the server will stop sending messages (including heartbeats) on each channel. This is normally at 5:00pm.

2.2.4 Error Recovery

2.2.4.1 System Component Failure

If a system component fails and requires a failover or restart, there will be a short interruption in multicast dissemination from either Line A or Line B. The system is deployed in an active-active configuration with Line A and Line B being generated independently and so line arbitration will allow the client to continue receiving messages – see section 4 for more information about recovery.

2.2.4.2 Disaster Recovery

In the unlikely event of a disaster recovery situation at the primary site, OMD-CC will be brought up at the disaster recovery (DR) site.

During the interruption, no data will be sent including heartbeats.

A Disaster Recovery (DR) Signal message indicating the DR status will be sent on its dedicated channel when OMD-CC is brought up – see section 3 for more information about the DR Signal message.

IP addresses and ports that have been provided for the disaster site's retransmission service should be used. *See Connectivity Guide for more details.*

2.3 TRADING SESSIONS

Please refer to the official websites of Shanghai Stock Exchange and Shenzhen Stock Exchange for detailed information of the trading sessions of their markets.

3. MESSAGE FORMATS

3.1 DATA TYPES

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.1	●	●

The following table lists all the data types used by OMD.

Format	Description
String	ASCII characters which are left aligned and padded with spaces, unless otherwise specified.
Uint8	8 bit unsigned integer.
Uint16	Little-Endian encoded 16 bit unsigned integer.
Uint32	Little-Endian encoded 32 bit unsigned integer.
Uint64	Little-Endian encoded 64 bit unsigned integer.
Int16	Little-Endian encoded 16 bit signed integer.
Int32	Little-Endian encoded 32 bit signed integer.
Int64	Little-Endian encoded 64 bit signed integer.
Binary	Unicode encoding used for Chinese characters.

3.1.1 Null Values

From time to time certain fields cannot be populated and specific values are used to represent null.

Format	Null representation (Hex 2's complement)
Int32	0x80000000
Int64	0x8000000000000000

3.1.2 Currency Values

See the ISO-4217 Currency Codes for a full list of possible data values. Currently the system uses only the currency codes 'CNY' for Chinese Renminbi. HKEX may add or delete currency code(s), whenever applicable, in the future.

3.1.3 Decimal Values

Decimal values are sent as integers. This is done for efficiency - for example, a price value sent as "12345" and with 3 decimal places should be interpreted as "12.345". See individual fields for number of decimal places used.

3.2 PACKET STRUCTURE

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.2	●	●

Multicast packets are structured into a common packet header followed by zero or more messages. Messages within a packet are laid out sequentially, one after another without any spaces between them.

Packet Header	Message 1	Message 2	...	Message n

The maximum length of a packet is 1500 bytes which includes the multicast headers, Packet Header and Messages.

The packet header provides information including the total packet length, the number of messages within the packet, the sequence number of the first message and a send timestamp.

A packet will only ever contain complete messages. A single message will never be fragmented across packets.

The format of each message within a packet will vary according to message type. However, regardless of the message type, each message will start with a two-byte message size (MsgSize) followed by a two-byte message type (MsgType). These are described in the following table.

Table 1: MsgSize and MsgType Fields

Field	Format	Len	Description
MsgSize	Uint16	2	Message length (including this field)
MsgType	Uint16	2	Type of message. The valid values for MsgType are below: Sequence Reset (100) Logon (101) Logon Response (102) Disaster Recovery Signal (105) Retransmission Request (201) Retransmission Response (202) Refresh Complete (203) Market Definition (610) Security Definition (611) Security Status (621) Top of Book (655) Statistics (660)

3.3 PACKET HEADER

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.3	●	●

All packets will begin with a common packet header.

Offset	Field	Format	Len	Description
0	PktSize	Uint16	2	Size of the packet (including this field)
2	MsgCount	Uint8	1	Number of messages included in the packet
3	Filler	String	1	
4	SeqNum	Uint32	4	Sequence number of the first message in the packet
8	SendTime	Uint64	8	The number of nanoseconds since *January 1*, 1970, *00:00:00 GMT*, precision is provided to the nearest millisecond.
Packet length			16	

3.4 CONTROL MESSAGES

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.4	●	●

3.4.1 Heartbeat

Heartbeats consist of a packet header with MsgCount set to 0. They do not carry a sequence number and therefore do not increment the sequence number of the multicast channel. SeqNum is set to the sequence number of the previous message sent on the channel.

The Heartbeat message will be identical for all the services.

3.4.2 Sequence Reset (100)

The Sequence Reset message is sent on each multicast channel at start of day.

The client must ignore the sequence number of the Sequence Reset message itself, and set the next expected sequence number to NewSeqNo. The client may receive multiple sequence reset messages from all channels. Whenever the Sequence Reset message is received, clients must clear all cached data for all eligible instruments traded in the SSE and SZSE and then subscribe to the refresh channels to receive the current state of the markets.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	100 Sequence Reset
4	NewSeqNo	Uint32	4	New sequence number.	Always set to 1
Total Length			8		

3.4.3 Disaster Recovery Signal (105)

The Disaster Recovery (DR) Signal message is sent on a dedicated multicast channel (DR channel) whenever site failover is triggered. In normal situation, the dedicated DR channel only carries Heartbeat till end of business day.

When site failover begins, DR Signal is sent with “DRStatus=1” indicating that the DR process has been activated. Clients should then clear all cached market data and prepare their own system for the site failover. When the site failover process finishes, DR Signal will be sent with “DRStatus=2” thereupon clients could start rebuild the latest market image from the refresh service. The same DR Signal will be sent periodically until end of business day.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	105 DR Message
4	DRStatus	Uint32	4	Status during site failover	1 – DR in progress 2 – DR completed
Total Length			8		

3.5 RETRANSMISSION

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.5	●	●

Refer to Retransmission service for details on the retransmission messages.

Note that when the Logon (101) or Retransmission Request (201) messages are sent to the OMD-CC server, the client must also include a packet header as shown below:

Offset	Field	Format	Len	Values	Notes
0	PktSize	Uint16	2	32	16 bytes for this header plus 16 bytes for either the Logon (101) or Retransmission Request (201) message
2	MsgCount	Uint8	1	1	One message only
3	Filler	String	1		Empty Filler
4	SeqNum	Uint32	4	0	The field is not used
8	SendTime	Uint64	8	0	The field is not used

After this header, the fields for either Logon (101) or Retransmission Request (201) should follow.

Also note that the same header is used by the RTS server when sending either Logon Response (102) or Retransmission Response (202) messages to clients. Again in this case the SeqNum and SendTime fields are not relevant and can be discarded.

3.5.1 Logon (101)

The Logon message enables client authentication. This is not required for multicast channels and is only used to for retransmission requests.

Normal operation: Client sends a Logon message containing username to the OMD-CC, which responds with a Logon Response message with the SessionStatus set to 0 (Session Active).

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	101 Logon
4	Username	String	12	Username to log on, padded with binary null characters	
Total Length			16		

3.5.2 Logon Response (102)

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	102 Logon Response
4	SessionStatus	Uint8	1	Status of the session	0 Session Active 5 Invalid username or IP Address 100 User already connected
5	Filler	String	3		
Total Length			8		

3.5.3 Retransmission Request (201)

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	201 Retransmission Request
4	ChannelID	Uint16	2	Multicast Channel ID with which the retransmission relates	
6	Filler	String	2		
8	BeginSeqNum	Uint32	4	Beginning of sequence	
12	EndSeqNum	Uint32	4	Message sequence number of last message in range to be resent	
Total Length			16		

3.5.4 Retransmission Response (202)

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	202 Retransmission Response
4	ChannelID	Uint16	2	Multicast Channel ID with which the retransmission relates	
6	RetransStatus	Uint8	1	Status of the Retransmission response	0 Request accepted 1 Unknown/Unauthorized channel ID 2 Messages not available 100 Exceeds maximum sequence range 101 Exceeds maximum requests in a day
7	Filler	String	1		
8	BeginSeqNum	Uint32	4	Beginning of sequence	
12	EndSeqNum	Uint32	4	Message sequence number of last message in range to be resent	
Total Length			16		

3.6 REFRESH

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.6	●	●

Refer to Refresh service for details on the Refresh Complete message.

3.6.1 Refresh Complete (203)

This message is published to mark the end of a refresh.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	203 Refresh Complete
4	LastSeqNum	Uint32	4	Sequence number with which the refresh is synchronized.	Numerical
Total Length			8		

3.7 REFERENCE DATA

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.7	●	●

3.7.1 Market Definition (610)

The Market Definition message is generated at the start of the business day for each market. (i.e. SSE or SZSE). Please note that Market Definition (610) messages for different markets are sent via different channels.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	610 Market Definition
4	MarketCode	String	4	Market code	ASHR SSE A-Share ASZR SZSE A-Share
8	MarketName	String	25	Market Name	Alphanumerical
33	CurrencyCode	String	3	Base currency code of the market.	CNY See also Currency Values in section 3.1.2 for full details.
36	NumberOfSecurities	Uint32	4	Number of securities within the market	
Total Length			40		

3.7.2 Security Definition (611)

This Security Definition message contains all the reference data for a security. Please note that Security Definition (611) messages for different markets (i.e. SSE and SZSE) are sent via different channels.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	611 Security Definition
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	6 digit security codes with possible values 1 – 999999
8	MarketCode	String	4	Market code	ASHR SSE A-Share ASZR SZSE A-Share
12	ISINCode	String	12	ISIN code of the security.	
24	InstrumentType	String	4	Instrument type of the security.	EQTY Equities TRST Trusts
28	Filler	String	2		
30	SecurityShortName	String	40	Security short name	
70	CurrencyCode	String	3	Security currency code of the market.	CNY See also Currency Values in section 3.1.2 for full details.
73	Filler	Binary	60		
133	SecurityNameGB	Binary	60	Security name in Simplified Chinese using Unicode	**Unicode UTF-16LE encoded**
193	LotSize	Uint32	4	Board lot size for the security	
197	PreviousClosingPrice	Int32	4	Previous closing price of the security	3 implied decimal places
201	Filler	String	1		
202	ShortsellFlag	String	1	Indicator for short-sell authorization.	Y Short-sell allowed N Short-sell not allowed
203	Filler	String	6		
209	ListingDate	Uint32	4	Date of security listing	The representation is YYYYMMDD Value is 19000101 for unknown listing date
213	Filler	String	7		
Total Length			220		

Note: PreviousClosingPrice may be set to 0, for example on the first day of listing (no existing previous closing price)

3.8 Status Data

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.8	●	●

3.8.1 Security Status (621)

The Security Status message is generated

- At the start of the business day if the security is halted.
- Whenever a security state or trading phase changes.

Please note that Security Status (621) messages for different markets (i.e. SSE and SZSE) are sent via different channels.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	621 Security Status
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	6 digit security codes with possible values 1 – 999999
8	SecurityTradingStatus	Uint8	1	Identifies the trading status of a security.	2 Trading Halt 3 Resume
9	Filler	String	3		
12	TradingPhaseCode	String	8	Identify the trading state of the security	Please refer to the TradingPhaseCode information in SSE and SZSE market data of this security for details.
Total Length			20		

3.9 PRICE DATA

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.9.1	●	●

3.9.1 Top Of Book (655)

The Top Of Book (TOB) message is generated when the top price level has been modified. Please note that Top of Book (655) messages for different markets (i.e. SSE and SZSE) are sent via different channels.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	655 Top of Book
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	6 digit security codes with possible values 1 – 999999
8	AggregateBidQuantity	Uint64	8	Aggregated number of shares on the ask side Provide Virtual Auction Volume during Call Auction period	
16	AggregateAskQuantity	Uint64	8	Aggregated number of shares on the bid side Provide Virtual Auction Volume during Call Auction period	
24	BidPrice	Int32	4	The bid price Provide Virtual Auction Price during Call Auction period	3 implied decimal places 0 means N/A
28	AskPrice	Int32	4	The ask price Provide Virtual Auction Price during Call Auction period	3 implied decimal places 0 means N/A
32	Filler	String	8		
Total Length			40		

3.10 VALUE ADDED DATA

3.10.1 Statistics (660)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD-CC SSE	OMD-CC SZSE
3.10.1	●	●

The Statistics message provides statistics including turnover. Please note that Statistics (660) messages for different markets (i.e. SSE and SZSE) are sent via different channels.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	660 Statistics
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	6 digit security codes with possible values 1 – 999999
8	SharesTraded	Uint64	8	Number of shares traded for a security	
16	Turnover	Int64	8	Current turnover	3 implied decimal places
24	HighPrice	Int32	4	Highest trade price currently performed for a security.	3 implied decimal places
28	LowPrice	Int32	4	Lowest trade price currently performed for a security	3 implied decimal places
32	LastPrice	Int32	4	Last trade price for a security and will be updated with Close price when Market Close for security with market code = ASHR (SSE A-Share)	3 implied decimal places
36	OpeningPrice	Int32	4	Opening price for a security	3 implied decimal places
40	Filler	String	12		
Total Length			52		

4. RECOVERY

OMD-CC provides three different mechanisms for recovering missed data:

- Line arbitration – using dual multicast channels (Line A and Line B)
- Retransmission Server – recovery of a limited number of messages
- Refresh Server – snapshot of current market state

These mechanisms should be used as described in the following table.

Table 2: Recovery Mechanisms

Event	Action
Packet lost on one either Line A or Line B	Try to recover data from the other line with a configurable timeout ("arbitration mechanism").
Dropped packet(s) on both Line A and Line B	Recover dropped message(s) from the Retransmission Server.
Late start up or extended intraday outage	Wait for a refresh of the current market state and then continue with real time messages.

4.1 GAP DETECTION

Each packet provides the sequence number (SN) of the first message it contains. This sequence number starts at 1 and increases with each subsequent message.

The sequence numbers provided in every packet header is calculated by adding the previous sequence number and the message count, as shown in table below:

Table 3: Sequence Number Calculation

Packet	Sequence Number	Message Count
Packet 1	1	4
Packet 2	5	2
Packet 3	7	1
Packet 4	8	3
Packet 5	11	1

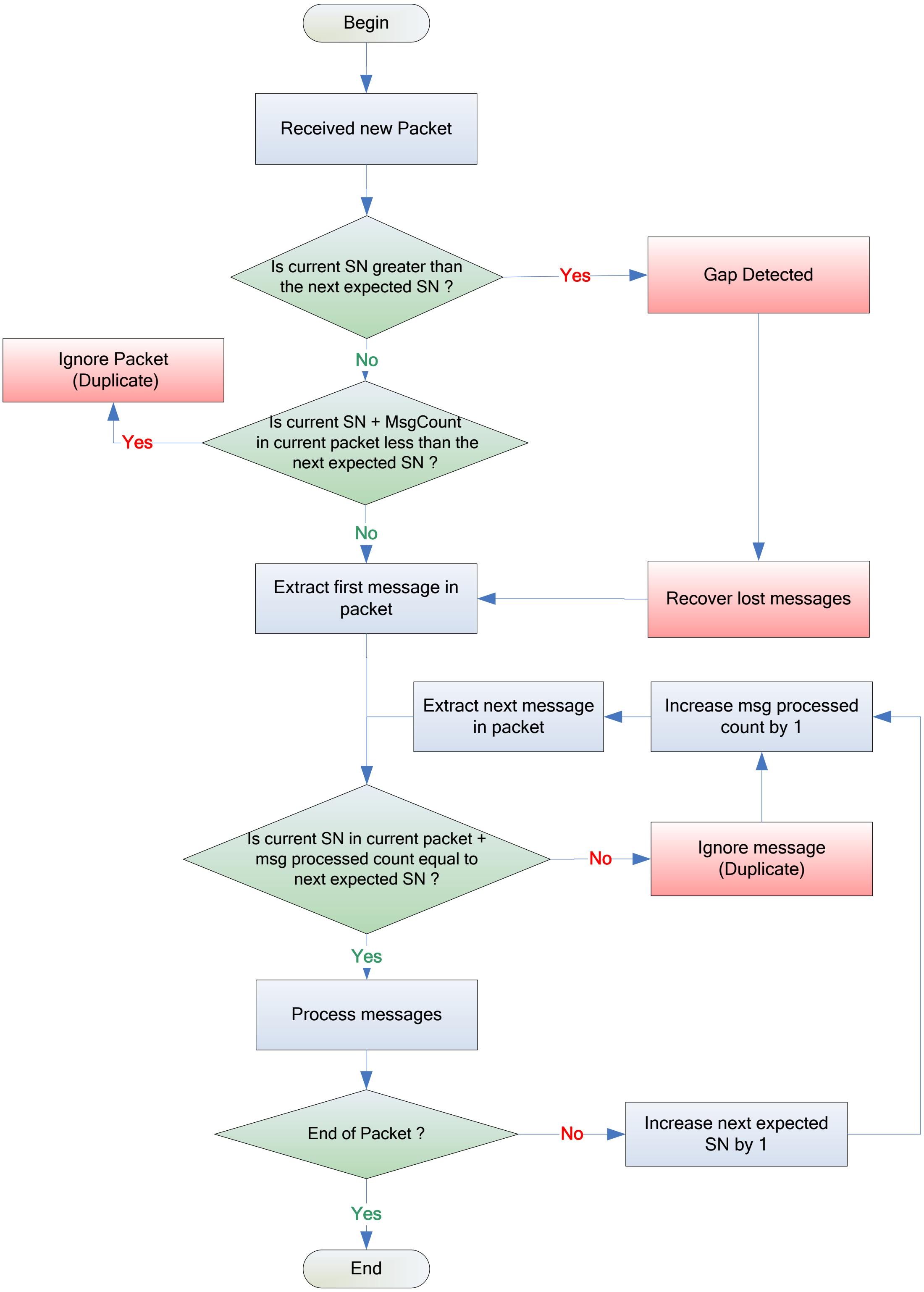
If the client drops the first five packets they would request a gap fill for messages 1-11.

All messages conform to the message level sequencing. Each channel has its own sequence number. This allows recipients to detect gaps or duplicates in each message sequence number and, if appropriate, reconcile them (line arbitration) with the primary or secondary multicast groups or request retransmission of the missing / corrupted messages.

Users should use this sequence number to detect gaps in the transmission of messages.

The following diagram illustrates how the message sequence number should be used to detect gaps in the feed.

Figure 2: Gap Detection using the Sequence Number (SN)


Begin
Received new Packet
Is current SN greater than the next expected SN ?
Yes
Gap Detected
No
Ignore Packet (Duplicate)
Yes
Is current SN + MsgCount in current packet less than the next expected SN ?
No
Extract first message in packet
Recover lost messages
Extract next message in packet
Increase msg processed count by 1
Is current SN in current packet + msg processed count equal to next expected SN ?
No
Ignore message (Duplicate)
Yes
Process messages
End of Packet ?
No
Increase next expected SN by 1
Yes
End

4.2 LINE ARBITRATION

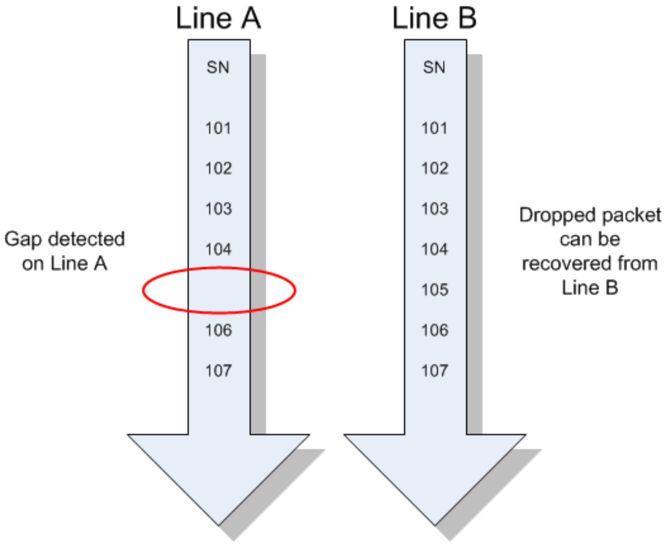
Client applications should check the sequence number (SN) and message count (MC) for every packet received. SNs are unique and increase monotonically for each service, the MC indicates the number of messages within each packet.

Line A and Line B are identical in terms of:

- SNs
- Messages that are sent
- Sequence in which messages are sent

However it is not guaranteed that a packet content between Line A and Line B will be the same. For example the third packet of the day from the Line A could contain SN 10 with MC 3, where as the third packet of the day from Line B could contain SN 9 with MC 4. For this reason clients must arbitrate on SN (at the message level) rather than packet content. Client applications should listen to both Line A and Line B in real-time. Clients should look at packets coming from both lines and process the ones that arrive first, regardless of whether they came from Line A or Line B. It is advisable to apply the “first come – first served” rule.

Figure 3 – Detecting Missing Packets



Additional Notes;

- The above example of a dropped packet is a simplified example assuming 1 message per packet, in reality each packet is likely to contain multiple messages
- Whilst the order of individual messages between Line A and Line B will be identical, there is no guarantee that the packets will contain exactly the same messages.
- In the example below, three packets are sent on each line, but message ‘TopOfBook3’ appears in one packet from Line A but in the subsequent packet on Line B.

Figure 4 – Normal Message Delivery

Primary				**Secondary**		
Messages	**MC**	**SN**		**SN**	**MC**	**Messages**
TopOfBook1 TopOfBook2 TopOfBook3	3	101		101	2	TopOfBook1 TopOfBook2
Statistics1 TopOfBook4	2	104		103	3	TopOfBook3 Statistics1 TopOfBook4
Statistics2 SecurityStatus1	2	106		106	2	Statistics2 SecurityStatus1

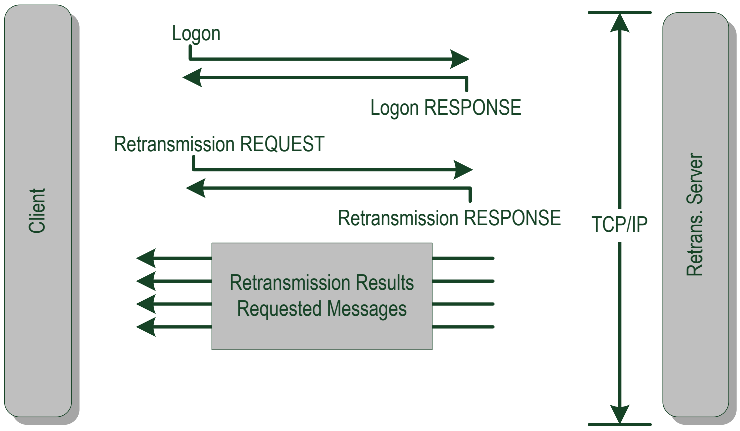
4.3 RETRANSMISSION SERVICE

The retransmission service is provided via the TCP/IP protocol and is designed to allow clients to recapture a small number of missed messages already published on the real time channels.

It is not intended that clients use the retransmission server to recover data after long outages or on late start up (in these situations, clients should use the Refresh service). To that end, it supports the retransmission of the last 50,000 messages per multicast channel only. The sequence range of messages that a client can request and the number of retransmission requests permitted per day is also limited.

The following diagram illustrates the message flow during a retransmission session:

Figure 5: Retransmission Request



Logon

The client establishes a TCP/IP connection and initiates a session by sending the Logon message. Once the client is authenticated the server will respond immediately with the Logon Response message. If the client does not send a Logon message within the logon timeout interval, the server will close the connection.

Logons may be rejected for the following reasons:

- Invalid username
- User already connected

In all cases the server will close the connection after sending the Logon Response message.

Making a request

The client can make a retransmission request by sending the Retrans Request message. The server will respond with a Retrans Response message to indicate whether the request has been accepted or not.

In the case of a successful request the server will send the requested messages immediately after the Retrans Response message.

The sequence numbers will be the same as when they were first sent on the real time multicast channel. The framing of the retransmitted messages into a packet may differ from the original transmission.

Retransmission requests may be rejected for the following reasons:

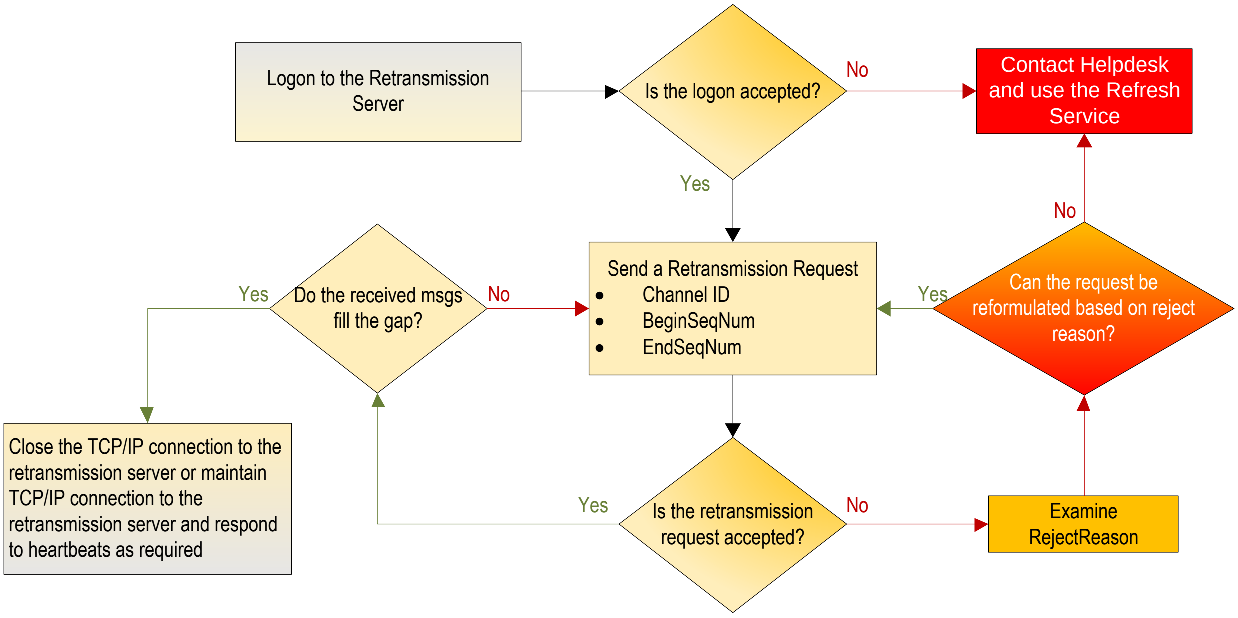
- Unknown channel ID or illegal (not authorized)
- Messages not available
- Exceeds maximum sequence range

- Exceeds maximum requests in a day

In the case where the client has exceeded the maximum number of requests allowed in a day, the server will close the connection after sending the Retrans Response message.

The following diagram is a guideline of the flow of logic when making a request:

Figure 6: Requesting Dropped Packets


Logon to the Retransmission Server
Is the logon accepted?
No
Contact Helpdesk and use the Refresh Service
Yes
No
Send a Retransmission Request
• Channel ID
• BeginSeqNum
• EndSeqNum
Yes
Can the request be reformulated based on reject reason?
Yes
Do the received msgs fill the gap?
No
Close the TCP/IP connection to the retransmission server or maintain TCP/IP connection to the retransmission server and respond to heartbeats as required
Yes
Is the retransmission request accepted?
No
Examine RejectReason

Multiple requests and concurrent sessions

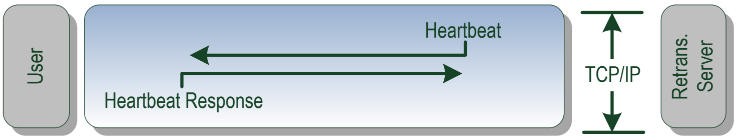
Clients can send multiple requests during a session and can keep the session open during idle periods by responding to heartbeats sent by the server. Concurrent sessions however will not be supported. Each user can only have one session open at a time.

If a client makes multiple requests, the server will process them serially. Clients are unable to cancel outstanding requests.

Heartbeats

To determine the health of the user connection on the TCP/IP channel, the Retransmission Server will send regular heartbeat packets to the user. The heartbeat frequency is 30 seconds. The client application must respond with a “Heartbeat Response” packet. The time out for this heartbeat response packet is set at 5 seconds. If no response is received by the server within this timeframe, the TCP/IP session will be disconnected.

Figure 7: Retransmission Server Heartbeat Message



A “heartbeat response” packet consists in an exact copy of the incoming heartbeat packet.

Closing the session

Sessions should be terminated by gracefully closing the TCP/IP connection.

System limits

The system limits mentioned above are set as follows:

System Limit	Value
Last number of messages available per channel ID	50,000
Maximum sequence range that can be requested	10,000
Maximum number of requests per day	1,000
Logon timeout (seconds)	5
Heartbeat interval (seconds)	30
Heartbeat response timeout (seconds)	5

Please note that the maximum number of requests per day limit is across all channels.

High availability

For each site, two sets of IP address and port are provided for the retransmission service in order to facilitate high availability. Clients may connect to both retransmission servers at the start of the day and maintain the connection during the day by responding to heartbeats.

In the event that Retransmission Server A (RTS A) does not respond to a logon or retransmission request, Retransmission Server B (RTS B) should be used.

In the event of a failure of either RTS A or RTS B, there may be a short period of unavailability. This failure should be detected by clients through the loss of connection. In this case the other RTS should be used.

RTS B should not be used as a means of requesting from two sources at the same time.

Disaster recovery

Two sets of backup IP address and port are also provided for the disaster site's retransmission service.

During normal conditions the retransmission service at the disaster site is not available. If clients attempt to connect, this will fail.

In the unlikely event of a disaster recovery situation, the retransmission service at the disaster site will be brought up and clients may connect via the backup IP addresses and ports.

4.4 REFRESH SERVICE

The refresh service is designed to allow clients to recover from a large scale data loss. This can happen after a late start or during a major outage.

Synchronization is on a per channel basis. For each real time multicast channel, there exists a corresponding refresh multicast channel on which snapshots of the market state are sent at regular intervals throughout the business day.

Market state

A snapshot of the market state is described in the table below.

Message	Snapshot description
Market Definition	Latest market static message for each market.
Security Definition	Latest security static message for each security.
Security Status	Security status message for halted securities, securities resumed trading with status changed to 'resumed' on the current trading day and SSE securities with the latest trading phase code.
Statistics	Latest statistics message for each security.

Message	Snapshot description
Top Of Book	Latest top of book message for each security

Refresh complete

A Refresh Complete message is sent at the end of a snapshot indicating the sequence number with which the snapshot is synchronized.

Snapshot processing

Below is an overview of the steps to carry out in order to process a channel snapshot.

- Subscribe to the real time multicast channel and cache received messages.
- Subscribe to the corresponding refresh multicast channel and discard messages until the Refresh Complete message is received.
- Process received messages until the next Refresh Complete message is received.
- Store the LastSeqNum sequence number provided in the Refresh Complete.
- Unsubscribe to the refresh multicast channel.
- Discard the cached real time messages with sequence number less than or equal to LastSeqNum.
- Process the remaining cached real-time messages and resume normal processing.

Missed messages

The retransmission server does not support refresh channels. If a client misses messages, it must wait for the next snapshot. Similarly if a client starts listening during the middle of a snapshot, it must wait for the next snapshot.