

NYSE OpenBook Ultra Client Specification

Version 1.7a

May 11, 2010

PREFACE

DOCUMENT HISTORY

Document Version	Date	Change Description
1.0	03/31/2008	Final version for distribution
1.1	04/23/2008	Added symbol mapping file layout
1.2	05/29/2008	Retrans Request IP Change, clarifications on repeating price points on Full and Delta update messages
1.3	05/30/2008	Description of MsgSize field in the Packet header Layout of Fields in PricePoints in Delta Message Layout of fields in the Book Refresh Request message (message type: 22) Description of ExtendedRefreshRequest message (msgtype 27) Data Types of fields in RetransResponse message (messagetype 10)
1.4	06/23/2008	"6 -End of Refresh Retransmission" (32,33) LinkFlag Description change (33,40) Symbol Index and Refresh Request clarification (46,48) LinkFlag Processing Notes (56) Multiple Packet refresh FAQ (58)
1.5	8/10/2009	UAT IP addresses added Msg types changed from 228/229 to 230/231 Refresh and Retrans IP's are now separated Full Update message change to Trading Status Delta Update message Change to Quote Conditions and Trading Status HeartbeatSubscription message type '19' added New reject reason code
1.6	10/16/2009	Correction the UAT IP address heading should say "Secondary UAT Refresh" not "Secondary UAT Retransmission"
1.7	11/25/2009	Updated following documentation review
1.7a	04/28/2010	Formatted into new template IP addresses removed and link to IP Addresses page added

ASSOCIATED DOCUMENTS

The following lists the associated documents, which either should be read in conjunction with this document or which provide other relevant information for the user:

- [SFTI US Technical Specification](#)
- [SFTI US Customer Guide](#)
- [NYSE Symbology](#)

CONTACT INFORMATION

For technical support please contact the Service Desk:

- Telephone: +1 212 383 3640 (International)
- Telephone: 866 873 7422 (Toll free, US only)
- Email: service.desk@nyx.com

FURTHER INFORMATION

For additional product information please visit, <http://www.nyxdata.com>

For updated capacity figures please visit our capacity pages at: <http://www.nyxdata.com/capacity>

For details of IP addresses, please visit our IP address pages at: <http://www.nyxdata.com/ipaddresses>

For a full glossary, see <http://www.nyxdata.com/glossary/>

TABLE OF CONTENTS

1	Introduction	6
2	NYSE OpenBook Feed Configuration	7
2.1	Multicast Groups	7
2.2	Joining Multicast Groups	7
2.3	Packet Retransmissions	8
2.4	Retransmission Request Thresholds	8
2.5	NYSE OpenBook Ultra Testing	8
2.6	NYSE OpenBook Ultra UAT Testing	9
3	NYSE OpenBook Ultra Operational Information	10
3.1	Publication Period	10
3.2	Gap Detection	10
3.3	Dual Site	10
4	NYSE OpenBook Ultra Message Specifications	11
4.1	Data Delivery Format	11
4.2	General Processing Notes	11
4.2.1	FullUpdate Processing Notes	11
4.2.2	DeltaUpdate Processing Notes	11
4.3	Sequence Numbers	11
4.4	Symbols	12
4.5	Symbol Mapping File Layout	12
4.6	Prices	13
4.7	NYSE OpenBook Ultra Data Messages	13
4.8	Message Header Format	13
4.9	Full Update Message Format	15
4.10	Delta Update Message Format	17
Appendix A	Common PDP Message Structure	20
A.1.	General Processing Notes	20
A.2.	Common Message Header Format	20
A.3.	Sequence Number Reset	22
A.4.	Heartbeat Subscription Message	22
A.5.	Heartbeat Messages	23
A.6.	Heartbeat Response Message	23
A.7.	Retransmission Request Message	24
A.8.	Symbol Index Mapping Request Message	25
A.9.	Book Refresh Request	25
A.10.	Extended Book Refresh Request	26
A.11.	Retransmission Response Message	27
A.12.	Retransmission Message	28
A.13.	Symbol Index Mapping Message	29
A.14.	Message Unavailable	30

Appendix B	Message Processing	32
B.1.	Processing of Messages	32
B.2.	Processing of Sequence Number Reset Messages	33
B.3.	Processing of Heartbeat Messages	34
B.4.	Processing of Heartbeat Response Messages	35
B.5.	Processing of Data Messages	36
B.6.	Processing of Gap Handling	37
B.7.	Processing of Line Level Retransmissions	38
B.8.	Processing of Refresh Messages	39

1 INTRODUCTION

NYSE Event Based OpenBook is a faster and enhanced version of the OpenBook Real-time product that will further increase the transparency of the NYSE market and provide data to customers as its available in the market. NYSE Event Based OpenBook will be available in addition to the current one-second update product available today. Event Based OpenBook is a data feed of NYSE limit-order book, as well as displayed crowd and DMM interest. Data is provided in a new format containing aggregate limit-order volume and individual event by event volume, action and price information for all bid and offer prices in all NYSE-traded issues updated as an event occurs between 7:30am and 4:00pm EST. A true real-time data feed that provides investors with the ability to see what the specialist sees.

2 NYSE OPENBOOK FEED CONFIGURATION

The following chapter provides connectivity information for the NYSE OpenBook Ultra data feed.

2.1 MULTICAST GROUPS

Due to capacity measures and data granularity requested by several customers, the NYSE data feed is split into twenty (20) data streams.

Each data stream will deliver a set of update messages for a certain range of symbols. The table below describes the categorization of the data feed.

Table 1 Multicast Groups

Feed Name	Description
OB_AA	Multicast Groups assigned to deliver book messages of symbols starting with letter A
OB_BB	Multicast Groups assigned to deliver book messages of symbols starting with letter B
OB_CC	Multicast Groups assigned to deliver book messages of symbols starting with letter C
OB_DD	Multicast Groups assigned to deliver book messages of symbols starting with letter D
OB_EE	Multicast Groups assigned to deliver book messages of symbols starting with letter E
OB_FF	Multicast Groups assigned to deliver book messages of symbols starting with letter F
OB_GG	Multicast Groups assigned to deliver book messages of symbols starting with letter G
OB_HH	Multicast Groups assigned to deliver book messages of symbols starting with letter H
OB_IJ	Multicast Groups assigned to deliver book messages of symbols starting with letters I or J
OB_KK	Multicast Groups assigned to deliver book messages of symbols starting with letter K
OB_LL	Multicast Groups assigned to deliver book messages of symbols starting with letter L
OB_MM	Multicast Groups assigned to deliver book messages of symbols starting with letter M
OB_NN	Multicast Groups assigned to deliver book messages of symbols starting with letter N
OB_OO	Multicast Groups assigned to deliver book messages of symbols starting with letter O
OB_PQ	Multicast Groups assigned to deliver book messages of symbols starting with letters P or Q
OB_RR	Multicast Groups assigned to deliver book messages of symbols starting with letter R
OB_SS	Multicast Groups assigned to deliver book messages of symbols starting with letter S
OB_TT	Multicast Groups assigned to deliver book messages of symbols starting with letter T
OB_UV	Multicast Groups assigned to deliver book messages of symbols starting with letters U or V
OB_WZ	Multicast Groups assigned to deliver book messages of symbols starting with letter W - Z

2.2 JOINING MULTICAST GROUPS

To access NYSE OpenBook Ultra, subscribers must join the multicast groups for primary feeds, as well as secondary feeds to assist in recovery. To request retransmissions of lost packets, subscribers must establish a TCP/IP connection (see [Packet Retransmissions](#)). Please refer to [Message Processing](#) for diagrams that illustrate message processing and retrieval.

Data feeds for specific stocks are sent to different multicast addresses. This addressing scheme allows customers to subscribe to the specific data feeds and channels they need. Data feeds types are:

- Multicast NYSE OpenBook Ultra (includes Symbol update messages)
- Multicast Retransmission

- Multicast Refresh (Request Based).

2.3 PACKET RETRANSMISSIONS

In the event a packet is lost on the primary feed for a multicast group, clients can retrieve the lost packet from the secondary feed. UDP can at times be unreliable and may drop packets from both the primary and secondary data feeds. If a packet is lost from both the primary and secondary feeds, clients then make a TCP/IP request to have the packets resent. Packets are resent via the Retransmission Multicast Feed.

Subscribers have the option to connect to the TCP/IP Recovery Server to request dropped packets from the OpenBook feed. This method is highly recommended in order to maintain a stable and accurate order book. The Recovery Server accepts connections on predefined addresses and ports and requires a heartbeat reply before responding to requests. It accepts primary and secondary connections to assist recovery on the subscriber's end.

After a client establishes a TCP/IP connection, NYSE OpenBook Ultra will send a heartbeat request message to the client. Clients should respond to this request with a heartbeat response within a specific timeframe. This timeframe is currently set to Sixty (60) seconds but is subject to change - so clients should make this configurable. (Clients will be informed of changes to the timeframe via customer notice.) Regardless of the timeframe, the client should respond immediately with a heartbeat response message. After receiving the initial heartbeat response, the Recovery Server will send heartbeats to the client every 60 seconds to ensure that the TCP/IP connection is live.

Note that the Source ID that the client specifies in the heartbeat response message will be validated by NYSE OpenBook Ultra. Each Source ID may only be logged in once per port at any given time.

To define a Source ID for retransmission purposes, contact the Service Desk and provide the desired Source ID. The Service Desk will evaluate and approve or reject Source IDs. If rejected, a new Source ID must be defined. If approved, the Service Desk will make the necessary updates on the NYSE OpenBook Ultra side to add the Source ID and applicable rules.

2.4 RETRANSMISSION REQUEST THRESHOLDS

The table below summarizes the Retransmission request thresholds for the NYSE OpenBook Ultra feed. The numbers below represent the thresholds per channel.

Table 2 Retransmission Request Thresholds

Capability	Description	Threshold	Action
Prevention of invalid subscribers	Incoming requests from subscribers that are not in the enabled subscriber's source ID list will not be honored. PDP subscribers will need a source ID, which is a string that uniquely identifies the subscriber of the retransmission requests. Please contact the Service Desk to get a unique source ID.	N/A	Request will not be processed.
Limitation of Requests for a large number of packets	Only retransmission requests for 1000 packets or less will be honored.	1000	Request will not be processed.
Limitation of Generic Requests Time Interval	If the generic request on a message which is not within this threshold, the request will not be honored.	75000	Request will not be honored.
Limitation of Generic Requests	If the number of a subscriber's generic requests reaches the threshold number of requests per day, the subscriber will be blocked and its retransmission request will no longer be honored during that particular day.	500	Subsequent retransmissions requests from that subscriber will be blocked.
Limitation of requests for refresh messages	Only refresh requests for 5000 messages or less will be honored.	5000	Request will not be honored.

2.5 NYSE OPENBOOK ULTRA TESTING

Replay tests are generally run at night (**Tuesday and Thursday from 7:00pm to 9:00pm**) and over different multicast groups than the production environment so that subscribers do not need to worry about incorrect data over the production lines.

The data replayed over this network is from a previous trading session (canned) - all messages or a range of messages for a given service in their original sequence.

2.6 NYSE OPENBOOK ULTRA UAT TESTING

Replay tests in the User Acceptance environment are generally run intra-day **Monday to Friday from 9:00am to 5:00pm** and over different multicast groups than the production environment and test environment so that subscribers do not need to worry about incorrect data over the production lines.

The data replayed over this network is from a previous trading session (canned) - all messages or a range of messages for a given service in their original sequence.

3 NYSE OPENBOOK ULTRA OPERATIONAL INFORMATION

3.1 PUBLICATION PERIOD

The following section specifies the frequency and publication period for each message type disseminated by the NYSE OpenBook Ultra product.

Table 3 Publication Period

Message	Message Type	Publication Period
NYSE OpenBook Ultra Full Update Message	230	An OpenBook Full update message is generated at the start of day, upon request, or when there is a failure on NYSE systems. The initial OpenBook Full update message will be transmitted at approximately 2:00am (EST)
NYSE OpenBook Ultra Delta Message	231	An OpenBook delta message is generated based on events. Every OpenBook delta message will be transmitted based on that event. The transmission time for the message is between 7:30am (EST) until market close (4:00pm (EST) for most securities). Please check the NYSE website for any changes to trading hours.

3.2 GAP DETECTION

The PDP Distribution System will assign all data packets a unique, sequential message number. This will allow recipients to identify 'gaps' in the message sequence and, if appropriate, reconcile them with the primary/secondary multicast groups or request retransmission of the missing/corrupted data packets.

For the NYSE OpenBook Ultra product, each data stream will have its unique set of sequence numbers. In other words, the message sequence for NYSE OpenBook Ultra A channel (OB_AA) is independent from the message sequence of D channel (OB_DD) and so on. The following is an example of sequencing for each channel.

OB_AA : SeqNum=1, SeqNum=2, SeqNum=3, ..., SeqNum=n.
 OB_BB : SeqNum=1, SeqNum=2, SeqNum=3, ..., SeqNum=x.
 OB_CC : SeqNum=1, SeqNum=2, SeqNum=3, ..., SeqNum=m.
 OB_DD: SeqNum=1, SeqNum=2, SeqNum=3, ..., SeqNum=y.

If there is a gap in the sequence of any channel, it has to be recovered independently. Therefore, if there is a sequence number gap on channel OB_DD, then the gap filling has to be done through the appropriate OB_DD retransmission channel. The same is valid for the other channels.

3.3 DUAL SITE

NYSE OpenBook Ultra data is redundant service with identical data transmitted through Primary and Secondary Multicast groups. This will allow customers to receive two redundant feeds. Additionally, any message can be retransmitted upon request.

4 NYSE OPENBOOK ULTRA MESSAGE SPECIFICATIONS

The NYSE OpenBook Ultra is a real-time full limit order depth of book information for all NYSE listed securities. Updates are sent as an event(s) occurs on the book. Data is enriched with value added information such as the change reason, the LinkID in the event of an execution, the number of orders at the current price point, as well as the source timestamp in microsecond granularity.

4.1 DATA DELIVERY FORMAT

The NYSE OpenBook Ultra service uses the push-based publishing model. This means that data will be published based on its availability. Once an order book change is available, it will be published to NYSE OpenBook Ultra Subscribers.

4.2 GENERAL PROCESSING NOTES

The following processing notes apply to the messages sent through the feed.

- All fields will be sent for every packet.
- Only field values will appear in the published messages (e.g. no names, 'tags', sizes will appear in the message). The field names that appear in the descriptions below are for reference purposes only.
- All the fields are contiguous, with reserved fields for alignment issues.
- All field sizes are fixed and constant.
- The source time referenced will be using Eastern Standard Time (EST).
- Binary fields are provided in big-endian format.
- ASCII string fields are left aligned and null padded.

The following processing notes apply to both the Full Update and Delta Update Message:

- Any Update Message can span multiple messages and/or multicast packets. This is because a maximum limit on the size of a packet to be 1500 bytes. This is done to avoid the "splitting" of messages due to network MTU restrictions.
- For Update Messages that span multiple messages/packets, all fields not part of the 'price point' section will be repeated for each packet. To determine the number of 'price point' in any given message, use the following formula:

$$\text{NumPricePoints} = (\text{SizeOfMessage} - \text{sum}(\text{size of fixed fields of message})) / \text{sum}(\text{size of fields for a price-point})$$

4.2.1 FullUpdate Processing Notes

Full Update Messages that span multiple packets/messages must be processed as one complete message.

For Full Update Messages that span multiple packets, if a packet is lost, then the whole message should be considered lost.

Full Update Messages contain all active price points regardless of prior period activity.

4.2.2 DeltaUpdate Processing Notes

Delta Update Messages that span multiple packets must be processed as one complete message.

For Delta Update Messages that span multiple packets, if a packet is lost, then the whole message should be considered lost.

All price points containing a zero (0) quantity should be removed as an active price point.

If no changes have occurred for a given symbol (e.g., an inactive stock) since the last publication, no Delta Update Message is generated for that symbol

4.3 SEQUENCE NUMBERS

All messages conform to the line level sequencing. Each channel A, B, C, D, etc has its own sequence number. Subscribers can use sequence numbers to determine the following:

- Missing (gapped) messages
- Unordered messages
- Duplicate messages.

Clients should note that the message sequence number per channel might restart from one following a failure recovery. A reset sequence number message will be sent to clients via the Multicast Groups to inform of such event.

4.4 SYMBOLS

The stock symbols represented in this feed include the root and optional suffix utilizing NYSE Symbology.

For example, if a symbol's root is "ABC" and its suffix is "PRA", the symbol's root/suffix will be represented as: "ABC PRA\0\0\0\0". Between the root and the suffix there will be one space. After the suffix, null values follow to fill the 11 characters allocated for the stock symbol field.

Note: "\0" represents a null value

To ensure high throughput and low latency, symbols are identified using a Symbol Index Mapping Table in the delta messages (msg type 229). This is an ordered list from 1 to N of all symbols per multicast group. Symbol Indices are unique for every symbol and do not change each trading day. New symbols are appended to the end of the symbol mapping index and symbol that removed do not have their index number reused.

The symbol mapping is available via four methods:

- NYSE OpenBook Ultra Full Update Message
- NYSE OpenBook Ultra Full Update Refresh Message
- Symbol Update Message
- FTP file on ftp.nysedata.com/OpenBook/SymbolMapping/SymbolMap.xml
LoginID: anonymous

Note: The symbol mapping file has both NYSE and Amex securities. Amex Securities are represented by Channel "AZ" and also have the exchange code "A"

4.5 SYMBOL MAPPING FILE LAYOUT

The layout below is for the symbol mapping xml file found on the ftp.nysedata.com servers. The XML file is updated twice daily – at 12am (EST) and 8:30am (EST)

```
<?xml version='1.0' encoding='utf-8'>
  <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema' >
    <xs:complexType name='SymbolMap' >
      <xs:sequence>
        <xs:element name='Symbol' type='xs:string' >
        </xs:element>
        <xs:element name='Index' type='xs:int' >
        </xs:element>
        <xs:element name='Channel' type='xs:int' >
        </xs:element>
        <xs:element name='ExchangeID' type='xs:int' >
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:element name='SymbolMappingFile' >
      <xs:complexType>
        <xs:sequence>
          <xs:element name='SymbolMap' type='SymbolMap' >
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

4.6 PRICES

Prices in this feed are represented by two fields, separating the denominator and the numerator. All prices in the feed share a common denominator, which is represented in the PriceScaleCode.

The PriceScaleCode field value represents the common denominator for the following formula:

$$Price = \frac{Numerator}{10^{PriceScaleCode}}$$

For example, a price of 27.56 is represented by a Numerator of 2756 and a PriceScaleCode equals to 2.

4.7 NYSE OPENBOOK ULTRA DATA MESSAGES

The following table contains a list of the message types in the NYSE OpenBook Ultra feed.

Table 4 NYSE OpenBook Ultra Data Messages

Message	Message Type	Publication Period
NYSE OpenBook Ultra Full Update Message	230	An OpenBook Full update message is generated at the start of day, upon request, or when there is a failure on NYSE systems. This message contains the complete Order book with all price points, an aggregated quantity at each price point and a symbol mapping.
NYSE OpenBook Ultra Delta Message	231	An OpenBook delta message is generated based on events that occur on book such as interest being added, executions, cancellations and interest routed to a different market. This message contains the aggregated quantity at the price point, the quantity of the event, the reason for the message (new order, cancel, execution), a LinkID (if an execution occurs), and the number of orders.

4.8 MESSAGE HEADER FORMAT

All messages are preceded by a standard header format. The table describes the header fields of a NYSE OpenBook Ultra message.

Table 5 Message Header Format

Field	Offset	Size (Bytes)	Format	Description
MsgSize	0	2	Binary Integer	This field indicates the size of the message body in bytes, excluding these two bytes: Sequence Number Reset – '18 Bytes' Heartbeat Message – '14 Bytes' Heartbeat Response Message – '34 Bytes' Message Unavailable – '22 Bytes' Retransmission Request Message – '42 Bytes' Retransmission Response Message – '42 Bytes' Refresh Request Message – '50 Bytes' Extended Refresh Request Message – '38 Bytes' Symbol Update Request Message – '36 Bytes' Symbol Update Message – '28 Bytes' OpenBook Full Update Message - 'at least 46 Bytes'

Field	Offset	Size (Bytes)	Format	Description
				OpenBook Delta Update Message - 'at least 34 Bytes'
MsgType	2	2	Binary Integer	This field identifies the type of message '1' – Sequence Number Reset '2' – Heartbeat Message '5' – Message Unavailable '10' – Retransmission Response message '19' – Heartbeat Subscription message '20' – Retransmission Request Message '22' – Refresh Request Message '24' – Heartbeat Response Message '27' – Extend Refresh Request '34' – Symbol Update Request '35' – Symbol Update Message '230' – OpenBook Full Update Message '231' – OpenBook Delta Update Message
MsgSeqNum	4	4	Binary Integer	This field contains the message sequence number assigned by PDP for each product. It is used for gap detection. Also known as Line Sequence Number (LSN).
SendTime	8	4	Binary Integer	This field specifies the time message was created by PDP. The number represents the number of milliseconds since midnight of the same day.
ProductID	12	1	Binary Integer	'115' is the product value used in the PDP header to identify the NYSE OpenBook Ultra feed
RetransFlag	13	1	Binary Integer	A flag that indicates whether this is an original, retransmitted, or 'replayed' message. Valid values include: '1' – Original message '2' – Retransmitted message '3' – Message Replay '4' – Retransmission of a 'replayed' message '5' – Refresh Retransmission '6' - End of Refresh Retransmission '129' – Test Original Message '130' – Retransmission of a test message '131' – Replay of a test message '132' – Retransmission Replay of a Test Message
NumBodyEntries	14	1	Binary Integer	The number of times the message body repeats in the message. For example, if the body consists of only a single field (named Volume) and the "NumBodyEntries" field is 2, this signifies there were two events for the price point and as a result the number of bytes in the message body will be 8
LinkFlag	15	1	Binary	Contains the sequence of this packet in set of packets for

Field	Offset	Size (Bytes)	Format	Description
			Integer	<p>a refresh retransmission.</p> <p>Used in conjunction with the RetransFlag, indicates whether there are other packets in the refresh retransmission and if so gives the sequence number of this packet in that series, allowing clients to handle out of order UDP packets.</p> <p>This field applies only to Refresh Retransmissions (messages with RetransFlag of either 5 or 6).</p> <p>For all other messages/RetransFlag's this field is always set to 0.</p>

4.9 FULL UPDATE MESSAGE FORMAT

The table below describes the body fields of an OpenBook Full Update message (**MsgType '230'**)

Table 6 Full Update Message Format

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Integer	<p>This field indicates the number of bytes in this message including this field.</p> <p>MsgSize=sum(fixed fields) + number of price Points*sum (price point fields for 1 price point)</p>
SecurityIndex	18	2	Binary Integer	<p>This field identifies the numerical representation of the symbol.</p>
SourceTime	20	4	Binary Integer	<p>This field specifies the time when the full update was generated in the order book. The number represents the number of milliseconds since midnight of the same day.</p> <p>Example:</p> <p>If the time is 13:12 56 seconds, 170 milliseconds and 30 microseconds</p> <p>This field will contain the value 47576170</p>
SourceTimeMicroSecs	24	2	Binary Integer	<p>This field indicates the number of microseconds that have elapsed within the millisecond the full update message was generated in the book.</p> <p>For example</p> <p>If the sourcetime is 13:12:56 secs, 170ms and 30microsecs</p> <p>This field will contain value 30</p>
SymbolSeqNum	26	4	Binary Integer	<p>This field contains the sequence number assigned by the source system to this message also known as the EventID. The sequence number is unique only to a given stock. Hence orders for two different stocks may share the same source sequence number.</p>
SourceSessionID	30	1	Binary Integer	<p>This field contains the source-session identifier. This number is incremented with every new source-session (restart, matching engine intra-day stock add/move, etc) during the day. Default value is "1". The SymbolSeqNums may restart at a lower</p>

Field Name	Offset	Size (Bytes)	Format	Description
				number with every new session to ensure unique sequence numbers.
Symbol	31	11	ASCII String	This field contains the full symbol in NYSE Symbology. A sequence of characters representing the symbol, padded with NULLs
PriceScaleCode	42	1	Binary Integer	The PriceScaleCode code for the price fields in this message. Represents the number of digits after the decimal place in the price. Example: - For a price of 12.1, the PriceScaleCode is 1. - For price 13, the code is 0
QuoteCondition	43	1	ASCII Character	This field contains the current quote condition for the symbol The quote condition shall be blank if no quote condition exists (example when the Book is fast). Valid Values: 'E' - Slow on the Bid due to LRP or GAP Quote 'F' - Slow on the Ask due to LRP or GAP Quote 'U' - Slow on the Bid and Ask due to LRP or GAP Quote 'W' - Slow Quote due to a Set Slow list on both the bid and offer sides
TradingStatus	44	1	ASCII Character	The current trading status of the equity. Valid Values: 'P' - Pre-Opening for messages sent before the stock is opened on a trade or quote 'O' - The stock has opened or re-opened 'C' - The stock was closed from the Closing template 'H' - The stock is halted during a trading halt and has not resumed
Filler	45	1	Binary Integer	This is a filler, reserved for future use
MPV	46	2	Binary Integer	This field contains the minimum price variation, also known as Tick, minimum amount by which prices can differ.

The following fields represent a price point and can repeat in a message:

A price-point consists of 4 items

- The Buy/Sell Side Indicator
- The actual price at the "side"
- the total quantity available at the price point/side combination
- the total number of orders at that price-point/side

Note: There maybe 0 price points in a message due to internal matching engine processing. If that is the situation, continue to process the message to maintain the correct sequence number.

Field Name	Offset	Size (Bytes)	Format	Description
To identify the number of price points in the message, use the formula: (MsgSize - sum(size of fixed fields)) / size of 1 price point				
PriceNumerator	48	4	Binary Integer	This field contains the price (numerator) of this price point. Note: The price is represented by the PriceScaleCode and the PriceNumerator. For example, a price of 12.1 has a "price numerator" of 121 and a scalecode of 1
Volume	52	4	Binary Integer	This field contains the total interest quantity at a price point
NumOrders	56	2	Binary Integer	This field contains the number of orders at the current price point
Side	58	1	ASCII Character	This field indicates the side of the order Buy/sell. Valid Values: 'B' – Buy 'S' – Sell
Filler	59	1	Binary Integer	This is a filler, reserved for future use

4.10 DELTA UPDATE MESSAGE FORMAT

The table below describes the body fields of an OpenBook Delta Update message (**MsgType '231'**).

Table 7 Delta Update Message Format

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Integer	This field indicates the number of bytes in this message including this field. MsgSize=sum(fixed fields) + number of price Points*sum (price point fields for 1 price point)
SecurityIndex	18	2	Binary Integer	This field identifies the numerical representation of the symbol.
SourceTime	20	4	Binary Integer	This field specifies the time when the full update was generated in the order book. The number represents the number of milliseconds since midnight of the same day. Example: If the time is 13:12 56 seconds, 170 milliseconds and 30 microseconds This field will contain the value 47576170
SourceTimeMicroSecs	24	2	Binary Integer	This field indicates the number of microseconds that have elapsed within the second the full update message was generated in the book. For example If the sourcetime is 13:12:56 secs, 170ms and 30microsecs This field will contain value 30 micros

Field Name	Offset	Size (Bytes)	Format	Description
SourceSeqNum	26	4	Binary Integer	This field contains the sequence number assigned by the source system to this message. The sequence number is unique only to a given stock. Hence orders for two different stocks may share the same source sequence number.
SourceSessionID	30	1	Binary Integer	This field contains the source-session identifier. This number is incremented with every new source-session (restart, matching engine intra-day stock add/move, etc) during the day. Default value is "1". The SymbolSeqNums may restart at a lower number with every new session to ensure unique sequence numbers.
QuoteCondition	31	1	ASCII Character	The current quote condition for the symbol The quote condition shall be blank if no quote condition exists (example when the Book is fast). Valid Values: 'E' - Slow on the Bid due to LRP or GAP Quote 'F' - Slow on the Ask due to LRP or GAP Quote 'U' - Slow on the Bid and Ask due to LRP or GAP Quote 'W' - Slow Quote due to a Set Slow list on both the bid and offer sides
TradingStatus	32	1	ASCII Character	The current trading status of the equity. Valid Values: 'P' - Pre-Opening for messages sent before the stock is opened on a trade or quote 'O' - The stock has opened or re-opened 'C' - The stock was closed from the Closing template 'H' - The stock is halted during a trading halt and has not resumed
PriceScaleCode	33	1	Binary Integer	The PriceScaleCode code for the price fields in this message. Represents the number of digits after the decimal place in the price. Example: - For a price of 12.1, the PriceScaleCode is 1. - For price 13, the code is 0

The following fields represent a price point and can repeat in a message:

A price-point consists of 6 items

- The Buy/Sell Side Indicator
- The price at the "side"
- the total quantity available at the price point/side combination
- The quantity of the event
- the total number of orders at that price-point/side
- the "change" reason i.e. the reason the price point was affected

Field Name	Offset	Size (Bytes)	Format	Description
<p>Note: There maybe 0 price points in a message due to internal matching engine processing. If that is the situation, continue to process the message to maintain the correct sequence number.</p> <p>To identify the number of price points in the message, use the formula:</p> $(\text{MsgSize} - \text{sum}(\text{size of fixed fields})) / \text{size of 1 price point}$				
PriceNumerator	34	4	Binary Integer	This field contains the price (numerator) of this price point. Note: The price is represented by the PriceScaleCode and the PriceNumerator. For example, a price of 12.1 has a "price numerator" of 121 and a Pricescalecode of 1
Volume	38	4	Binary Integer	This field contains the total interest quantity at a price point
ChgQty	42	4	Binary Integer	The volume of the event taking place (i.e size of the order, cancel or execution)
NumOrders	46	2	Binary Integer	This field contains the number of orders at the current price point
Side	48	1	ASCII Character	This field indicates the side of the order Buy/Sell. Valid Values: 'B' – Buy 'S' – Sell
ReasonCode	49	1	ASCII Character	This field identifies why the volume at the price point was modified Valid Values: 'O' - New order/additional interest added 'C' - Cancel 'E' - Execution 'X' - Multiple events
LinkID1	50	4	Binary Integer	The LinkID identifies a unique transaction in the matching and allows you to correlate execution reports and quotes to the last sale. This field is populated only when an execution occurs. (Optional)
LinkID2	54	4	Binary Integer	The LinkID identifies a unique transaction in the matching and allows you to correlate execution reports and quotes to the last sale. This field is populated only when a repeat execution occurs at the same price point due to a single event (Optional)
LinkID3	58	4	Binary Integer	The LinkID identifies a unique transaction in the matching and allows you to correlate execution reports and quotes to the last sale. This field is populated only when a repeat execution occurs at the same price point due to a single event (Optional)

APPENDIX A COMMON PDP MESSAGE STRUCTURE

In broad terms, there are two types of messages transmitted as part of this protocol:

- Control - Control messages do not contain data per se; rather, they allow conversing parties to exchange session-specific information (e.g. 'reset sequence number').
- Data - Data messages are product specific and although they will adhere to the general specification, they are defined specifically in a later section.

A.1. General Processing Notes

The following processing notes apply to the Control and Data messages:

- All fields will be sent for every packet.
- Only field values will appear in the published messages (e.g. no names, 'tags', sizes will appear in the message). The field names that appear in the descriptions below are for reference purposes only.
- All the fields are contiguous, i.e. there is no explicit (or implicit) 'padding' between fields regardless of the juxtaposed data types, sizes, and alignment issues.
- All field sizes are fixed and constant.
- The source time referenced will be using Eastern Standard Time (EST).
- Binary fields are provided in big-endian format.
- All binary fields will be unsigned (unless otherwise specified)
- ASCII string fields are left align, null padded.

A.2. Common Message Header Format

All PDP messages will contain a Common Message Header with the exception of the Order Book Refresh Message. This model is akin to that of an envelope/letter paradigm. The message header comprises envelope information; the message body comprises the letter. All correspondence will use the same envelope format, regardless of content.

The intent of this design is to minimize development burden on behalf of Subscribers. That is, all Subscribers may implement line-level protocol processing once, and then need only develop parsing algorithms for messages of choice.

Table 8 Common Message Header Format

Field	Offset	Size (Bytes)	Format	Description
MsgSize	0	2	Binary Integer	This field indicates the size of the message body in bytes, excluding these two bytes: Sequence Number Reset – '18 Bytes' Heartbeat Message – '14 Bytes' Heartbeat Response Message – '34 Bytes' Message Unavailable – '22 Bytes' Retransmission Request Message – '42 Bytes' Retransmission Response Message – '42 Bytes' Refresh Request Message – '50 Bytes' Extended Refresh Request Message – '38 Bytes' Symbol Update Request Message – '36 Bytes' Symbol Update Message – '28 Bytes' OpenBook Full Update Message - 'at least 46 Bytes'

Field	Offset	Size (Bytes)	Format	Description
				OpenBook Delta Update Message - 'at least 34 Bytes'
MsgType	2	2	Binary Integer	This field identifies the type of message '1' – Sequence Number Reset '2' – Heartbeat Message '5' – Message Unavailable '10' – Retransmission Response message '19' – Heartbeat Subscription message '20' – Retransmission Request Message '22' – Refresh Request Message '24' – Heartbeat Response Message '27' – Extend Refresh Request '34' – Symbol Update Request '35' – Symbol Update Message '230' – OpenBook Full Update Message '231' – OpenBook Delta Update Message
MsgSeqNum	4	4	Binary Integer	This field contains the message sequence number assigned by PDP for each product. It is used for gap detection. Also known as Line Sequence Number (LSN).
SendTime	8	4	Binary Integer	This field specifies the time message was created by PDP. The number represents the number of milliseconds since midnight of the same day.
ProductID	12	1	Binary Integer	'115' is the product value used in the PDP header to identify the OpenBook Ultra feed
RetransFlag	13	1	Binary Integer	A flag that indicates whether this is an original, retransmitted, or 'replayed' message. Valid values include: '1' – Original message '2' – Retransmitted message '3' – Message Replay '4' – Retransmission of a 'replayed' message '5' – Refresh Retransmission '6' - End of Refresh Retransmission '129' – Test Original Message '130' – Retransmission of a test message '131' – Replay of a test message '132' – Retransmission Replay of a Test Message
NumBodyEntries	14	1	Binary Integer	The number of times the message body repeats in the message. For example, if the body consists of a field (named Volume) and the "NumBodyEntries" field is 2, the number of bytes in the message body will be 8
LinkFlag	15	1	Binary Integer	Contains the sequence of this packet in set of packets for a refresh retransmission. Used in conjunction with the RetransFlag, indicates

Field	Offset	Size (Bytes)	Format	Description
				<p>whether there are other packets in the refresh retransmission and if so gives the sequence number of this packet in that series, allowing clients to handle out of order UDP packets.</p> <p>This field applies only to Refresh Retransmissions (messages with RetransFlag of either 5 or 6).</p> <p>For all other messages/RetransFlag's this field is always set to 0.</p>

A.3. Sequence Number Reset

This message is sent to 'reset' the Sequence Number at start of day, in response to failures, etc. Note that this message will contain a valid sequence number. The message format is shown below.

Table 9 Sequence Number Reset Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Sequence Number Reset Message					
MsgSize	0	2	Binary Integer	18	See Common Message Header Format
MsgType	2	2	Binary Integer	'1'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the Sequence Number Reset Message					
NextSeqNumber	16	4	Binary Integer		This field contains the sequence number value that the recipient should expect in the immediately succeeding data packet. Note that this message will contain its own valid sequence number in the header portion of the message.

A.3.1 Sequence Number Processing Notes

Sequence numbers normally begin at one (1) and increase monotonically with each subsequent message. There are two scenarios where the sequence number is reset (besides the start of day). First, if the value should exceed the maximum value that the SeqNum field may contain, it will be reset to one (1).

Second, if PDP_OB has a failure and it recovers, it sends a sequence number reset message. The SeqNum field of that message will be set to one (1) and the NextSeqNumber field will be set to two (2), see [Processing of Sequence Number Reset Messages](#) for a suggested way of processing.

A.4. Heartbeat Subscription Message

Subscribers can optionally subscribe with the retrans/refresh server to receive heartbeats, by issuing this message with a valid sourceID. Once subscribed, the retrans/refresh server will forward PDP Heartbeats.

Note: Clients that do not explicitly subscribe for heartbeats would start receiving heartbeats only after they have issued at least 1 refresh/retrans request to the retrans/refresh server.

Table 10 Heartbeat Subscription Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Hearbeat Response Message					
MsgSize	0	2	Binary Integer	'34'	See Common Message Header Format
MsgType	2	2	Binary Integer	'19'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the Hearbeat Response Message					
SourceID	16	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned

A.5. Heartbeat Messages

Subscribers that choose to establish and remain connected to the TCP/IP Retrans/Refresh server will receive heartbeat message to let them know that the connection is still alive.

Table 11 Heartbeat Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Hearbeat Message					
MsgSize	0	2	Binary Integer	'14'	See Common Message Header Format
MsgType	2	2	Binary Integer	'2'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'0'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format

A.5.1 Heartbeat Message Processing Notes

Heartbeat messages will be sent with the same sequence number as the most recent message that was sent.

Heartbeat messages will only contain the PDP Message Header with an empty body.

Subscribers must respond to these heartbeat requests with a heartbeat message.

Please see [Processing of Heartbeat Messages](#) for a suggested way of processing.

A.6. Heartbeat Response Message

This message will be sent by subscribers that choose to establish and remain connected to the TCP/IP retransmission/refresh server intraday. This message lets NYSE know that the connection is still alive. Subscribers must respond to these heartbeat requests with a heartbeat response message.

Table 12 Heartbeat Response Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Heartbeat Response Message					
MsgSize	0	2	Binary Integer	'34'	See Common Message Header Format
MsgType	2	2	Binary Integer	'24'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the Heartbeat Response Message					
SourceID	16	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned

A.7. Retransmission Request Message

This message is sent by Subscribers requesting missing messages. The Message Archive and Retransmission (MART) will retransmit the appropriate message(s).

Table 13 Retransmission Request Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Generic Retransmission Request Message					
MsgSize	0	2	Binary Integer	'42'	See Common Message Header Format
MsgType	2	2	Binary Integer	'20'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the Generic Retransmission Request Message					
BeginSeqNum	16	4	Binary Integer		The beginning sequence number of the requested range of messages to be retransmitted.
EndSeqNum	20	4	Binary Integer		The end sequence number of the requested range of messages to be retransmitted.

Field Name	Offset	Size (Bytes)	Format	Value	Description
SourceID	24	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned

A.8. Symbol Index Mapping Request Message

This message is sent by Subscribers requesting the Symbol index mapping.

Table 14 Symbol Index Mapping Request Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Symbol Index Mapping Request Message					
MsgSize	0	2	Binary Integer	'36'	See Common Message Header Format
MsgType	2	2	Binary Integer	'34'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the Symbol Index Mapping Request Message					
SourceID	16	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned
SecurityIndex	36	2	Binary Integer		This field identifies the numerical representation of the symbol. SecurityIndex value can be zero, which is to request all symbol mapping for the multicast group.

A.9. Book Refresh Request

This message is sent by Subscribers requesting a full book refresh message.

Table 15 Book Refresh Request Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the NYSE Packet Unavailable Message					
MsgSize	0	2	Binary Integer	'50'	See Common Message Header Format
MsgType	2	2	Binary Integer	'22'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer		See Common Message Header Format
Defined below are the 'body' fields of the NYSE Packet Unavailable Message					
Symbol	16	16	ASCII String		A sequence of characters representing the symbol, padded with NULLs. The symbol contains the root, optionally followed by a space and an optional suffix in host format for e.g.: "IBM PRA\0\0\0\0"
SourceID	32	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned

A.9.1 Refresh Processing Notes

The refresh replies from MART may lag behind the current state disseminated by the OpenBook process. Thus, it is possible clients may not be able to correctly apply delta messages received from the main feed on the refresh from MART.

To prevent this, it is recommended that clients implement the following logic in the order listed below:

1. Join the main data multicast feed and queue up data for symbols for which refresh is required.
2. Next, join the MART retransmission channel and issue a refresh request for the desired symbol(s)
3. Read and process the resulting refresh retransmission entirely
4. Now, start processing the queued up data from main feed as follows:
 - a. If the EventId of the main data feed is lower or equal to that received from MART, discard the message(s)
 - b. If the EventId is exactly one more than that from MART, process/apply the messages.
 - c. If there was a gap in event ids between MART refresh message and that in main queue, re-request for a refresh from MART (go to step 2).

A.10. Extended Book Refresh Request

This is an extended form of a refresh request (MsgType 22) from clients to PDP.

The main differences are:

- Clients need to specify the Security Index in place of the actual symbol name (a SecurityIndex of 0 signifies all symbols. Note: Request for SecurityIndex 0 may be limited or blocked to prevent denial-of-service attacks)
- Clients may optionally request for a refresh of just a particular message group/type of the entire state. Currently, this field is not applicable and maybe set to 0 or 228.

Example:

If Type is set to 228, it indicates clients want a refresh of just the order state If Type is set to 0, it indicates clients want a complete refresh (i.e. all the states, if applicable).

Please note: When requesting a Symbol index Mapping or refresh request, you need to send to the request to symbols corresponding channel. i.e. symbol ABC to Channel AA, BBB to Channel BB otherwise your request will not be honored.

Table 16 Extended Book Refresh Request Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the NYSE Packet Unavailable Message					
MsgSize	0	2	Binary Integer	'38'	See Common Message Header Format
MsgType	2	2	Binary Integer	'27'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer		See Common Message Header Format
Defined below are the 'body' fields of the NYSE Packet Unavailable Message					
SourceID	16	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned
SecurityIndex	36	2	Binary Integer		This field identifies the numerical representation of the symbol. SecurityIndex value can be zero, which is to request all symbol mapping for the multicast group.
MsgType	38	2	Binary Integer		The group or type of refresh requested A value of 0 indicates 'all'

A.11. Retransmission Response Message

This message will be sent immediately via TCP/IP in response to the subscribers request for retransmission messages, i.e. Retransmission, refresh, Symbol Index Mapping. This message does not contain any information just the acceptance or rejection of the request message.

Table 17 Retransmission Response Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the NYSE Retransmission Request Message					
MsgSize	0	2	Binary Integer	'42'	See Common Message Header Format
MsgType	2	2	Binary Integer	'10'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer	'0'	See Common Message Header Format
Defined below are the 'body' fields of the NYSE Retransmission Request Message					
SourceSeqNum	16	4	Binary Integer		This field contains the request message

Field Name	Offset	Size (Bytes)	Format	Value	Description
					sequence number assigned by the client. It is used by the client to couple the request with the response message.
SourceID	24	20	ASCII String		This field represents the name of the source requesting retransmission. This field is null padded, left aligned
Status	40	1	Character		This is a flag that indicates whether the retransmissions request was accepted or rejected. Valid values: 'A' – Accepted 'R' – Rejected
RejectReason	41	1	Character		This is a flag that indicates the reason why the request was rejected. Valid values: '0' – Message was accepted '1' – Rejected due to permissions '2' – Rejected due to invalid sequence range '3' – Rejected due to maximum sequence range (>1000) '4' – Rejected due to maximum request in a day '5' – Rejected due to maximum number of refresh requests in a day '6' - Rejected. Request message seqnum TTL (Time to live) is too old. Use refresh to recover current state if necessary.
Filler	42	2	ASCII String		This is a filler, reserved for future use.

A.12. Retransmission Message

Upon receipt of a valid retransmission request message, the requested message(s) will be sent. This message(s) has the same message format and content as the original messages sent by the PDP_OB, with the exception that the 'RetransFlag' in the header is set to the value of '2' or '5' depending on whether the retransmission is for a non-replay or a replay retransmission message, respectively.

Table 18 Retransmission Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Retransmitted Message					
MsgSize	0	2	Binary Integer		See Common Message Header Format
MsgType	2	2	Binary Integer		It will be the MsgType of the original message sent by the PDP_OB.
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'2' or '5'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	Same as original message	See Common Message Header Format
LinkFlag	15	1	Binary Integer		See Common Message Header Format

All the 'body' fields of the Retransmitted Message are the same as the original message

A.13. Symbol Index Mapping Message

This message is sent by the NYSE in response to a Symbol Index Request or sent automatically when there are intraday symbol additions via Multicast.

Please note: When requesting a Symbol index Mapping, you need to send to the request to symbols corresponding channel. i.e symbol ABC to Channel AA, BBB to Channel BB otherwise your request will not be honored.

Table 19 Symbol Index Mapping Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the Symbol Index Mapping Request Message					
MsgSize	0	2	Binary Integer	'28'	See Common Message Header Format
MsgType	2	2	Binary Integer	'35'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer		See Common Message Header Format
Defined below are the 'body' fields of the Symbol Index Mapping Request Message					
Symbol	16	11	ASCII String		A sequence of characters representing the symbol, padded with NULLs. The symbol contains the root, optionally followed by a space and an optional suffix in host format for e.g.: "IBM PRA\0\0\0\0"
FILLER	27	1	ASCII String		This is a filler, reserved for future use.
SecurityIndex	36	2	Binary Integer		This field identifies the numerical representation of the symbol. SecurityIndex value can be zero, which

Field Name	Offset	Size (Bytes)	Format	Value	Description
					is to request all symbol mapping for the multicast group.

A.13.1 Retransmission Message Processing Notes

All Subscribers will receive retransmission messages through the retransmission channel.

Due to the multicast nature, subscribers will receive 'all' retransmission messages, including messages that were not requested by them.

Note that when a message for a particular symbol is retransmitted, a new message **for the same symbol** may be sent through the regular channel. This scenario is very likely to occur with busy symbols and may cause confusion as to which message contains the latest information on that symbol.

In order to resolve the conflict, the following qualification method should be applied:

- Check the MsgSeqNum field. A retransmitted message retains the same sequence number as the original message. Even refreshes are retransmitted with the original sequence numbers for the message they belonged to.
- The most current sequence number (SEQNUM) contains the latest information.
- If the SEQNUMS are the same: messages are the same, any of the two messages contains the same information.

Please see [Processing of Line Level Retransmissions](#) for a suggested way of processing.

A.14. Message Unavailable

This message will be sent to inform the subscribers of unavailability of a range of messages for which they may have requested retransmission via the Retransmission Multicast channels.

Table 20 Message Unavailable Message Format

Field Name	Offset	Size (Bytes)	Format	Value	Description
Set forth below are the 'header' fields of the NYSE Packet Unavailable Message					
MsgSize	0	2	Binary Integer	'22'	See Common Message Header Format
MsgType	2	2	Binary Integer	'5'	See Common Message Header Format
MsgSeqNum	4	4	Binary Integer		See Common Message Header Format
SendTime	8	4	Binary Integer		See Common Message Header Format
ProductID	12	1	Binary Integer	'115'	See Common Message Header Format
RetransFlag	13	1	Binary Integer	'1'	See Common Message Header Format
NumBodyEntries	14	1	Binary Integer	'1'	See Common Message Header Format
LinkFlag	15	1	Binary Integer		See Common Message Header Format
Defined below are the 'body' fields of the NYSE Packet Unavailable Message					
BeginSeqNum	16	4	Binary Integer		The beginning sequence number of the requested range of messages to be

Field Name	Offset	Size (Bytes)	Format	Value	Description
					retransmitted.
EndSeqNum	20	4	Binary Integer		The end sequence number of the requested range of messages to be retransmitted.

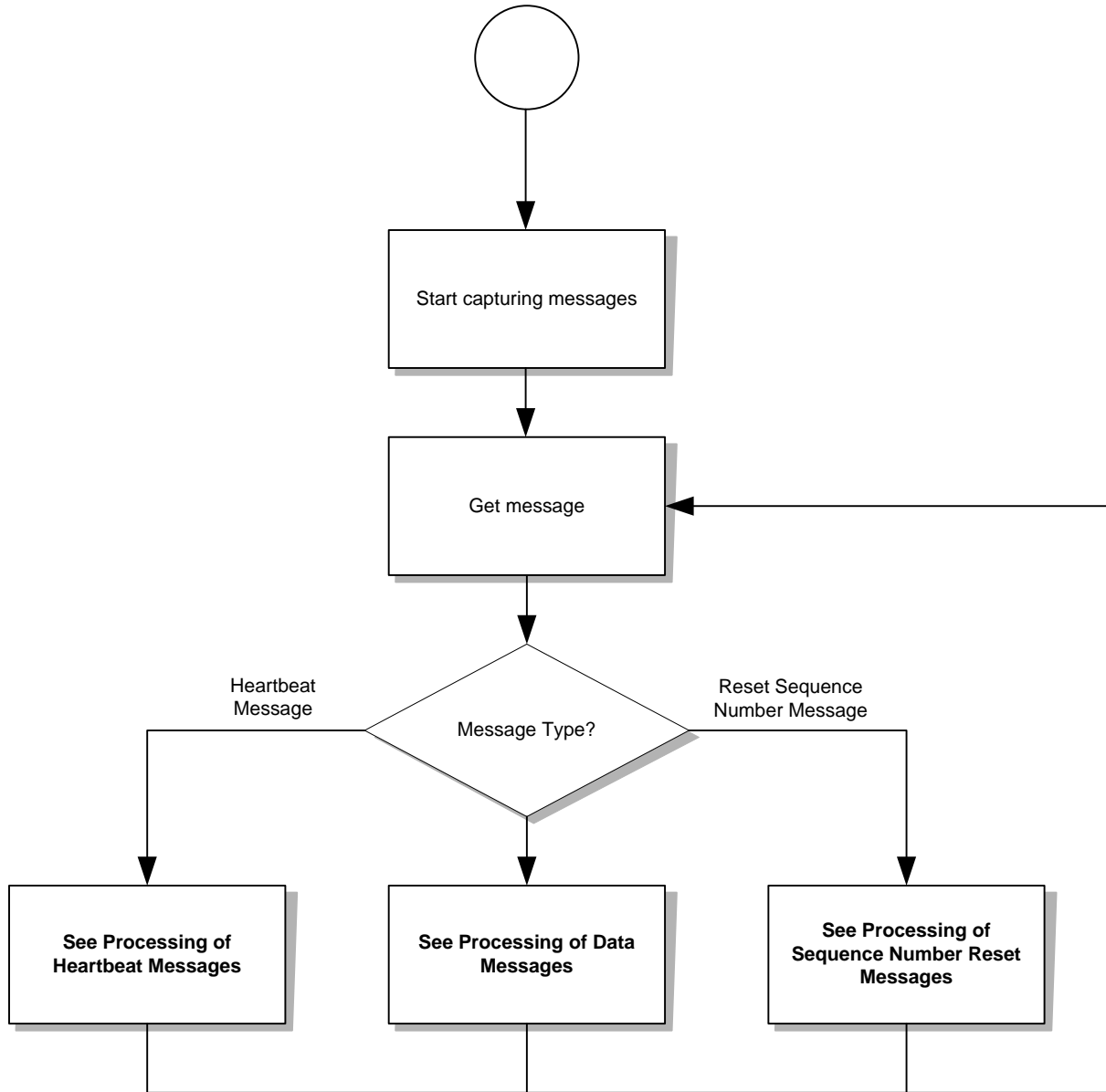
APPENDIX B MESSAGE PROCESSING

The following workflow diagrams simplify how the NYSE Book messages should be processed.

B.1. Processing of Messages

The following is the recommended way of processing messages:

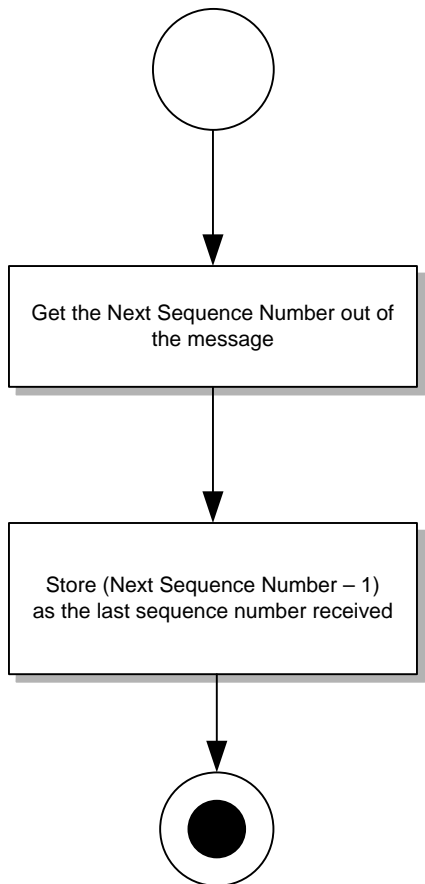
Figure 1 Message Processing



B.2. Processing of Sequence Number Reset Messages

The following is the recommended way of processing Sequence Number Reset messages:

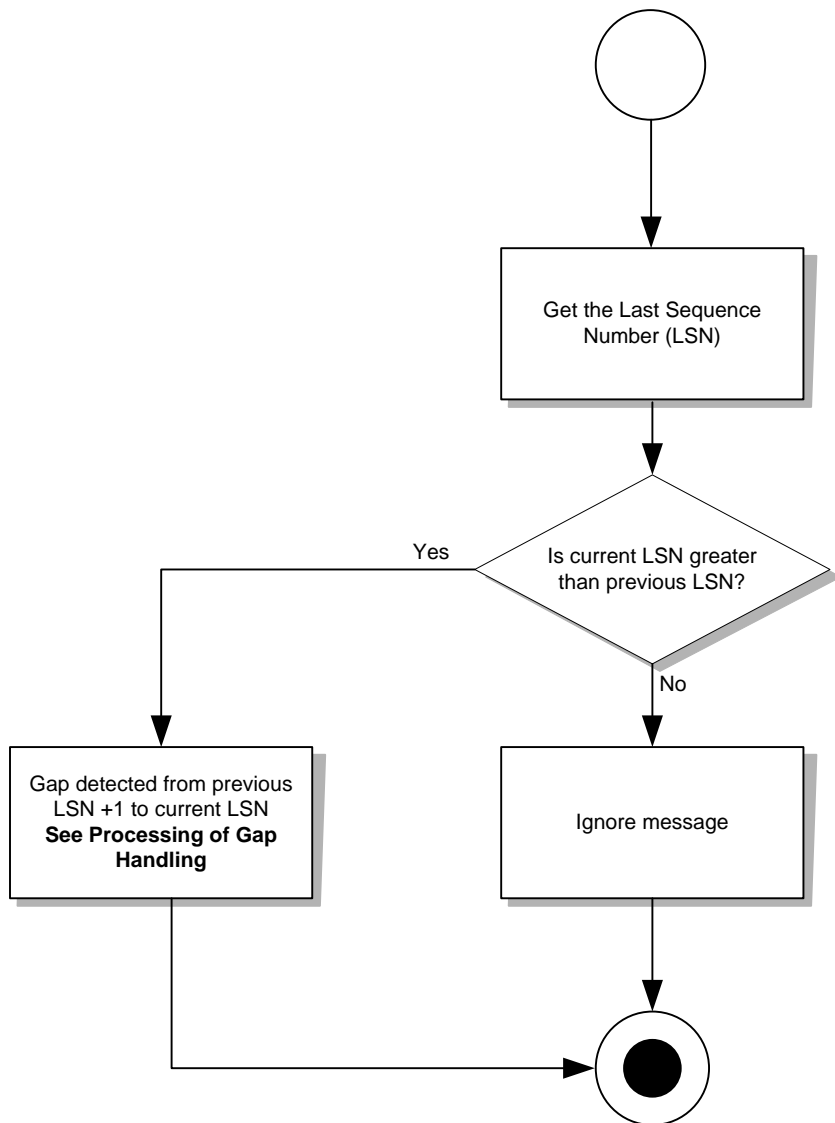
Figure 2 Sequence Number Reset Message Processing



B.3. Processing of Heartbeat Messages

The following is the recommended way of processing Heartbeat messages:

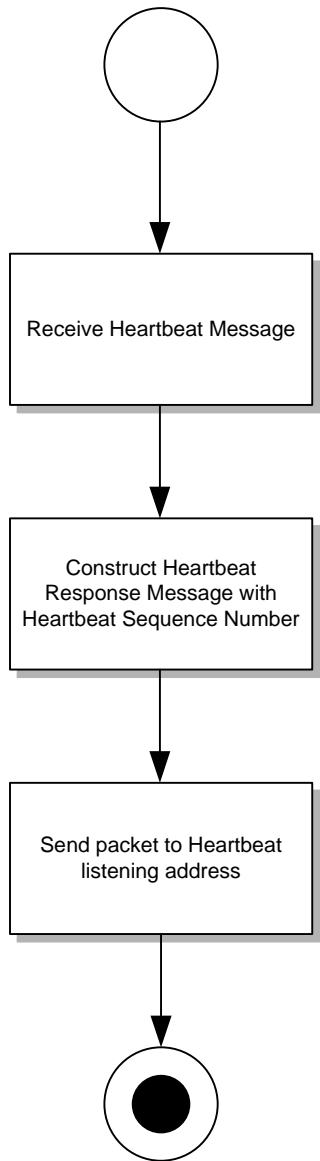
Figure 3 Heartbeat Message Processing



B.4. Processing of Heartbeat Response Messages

The following is the recommended way of processing Heartbeat messages:

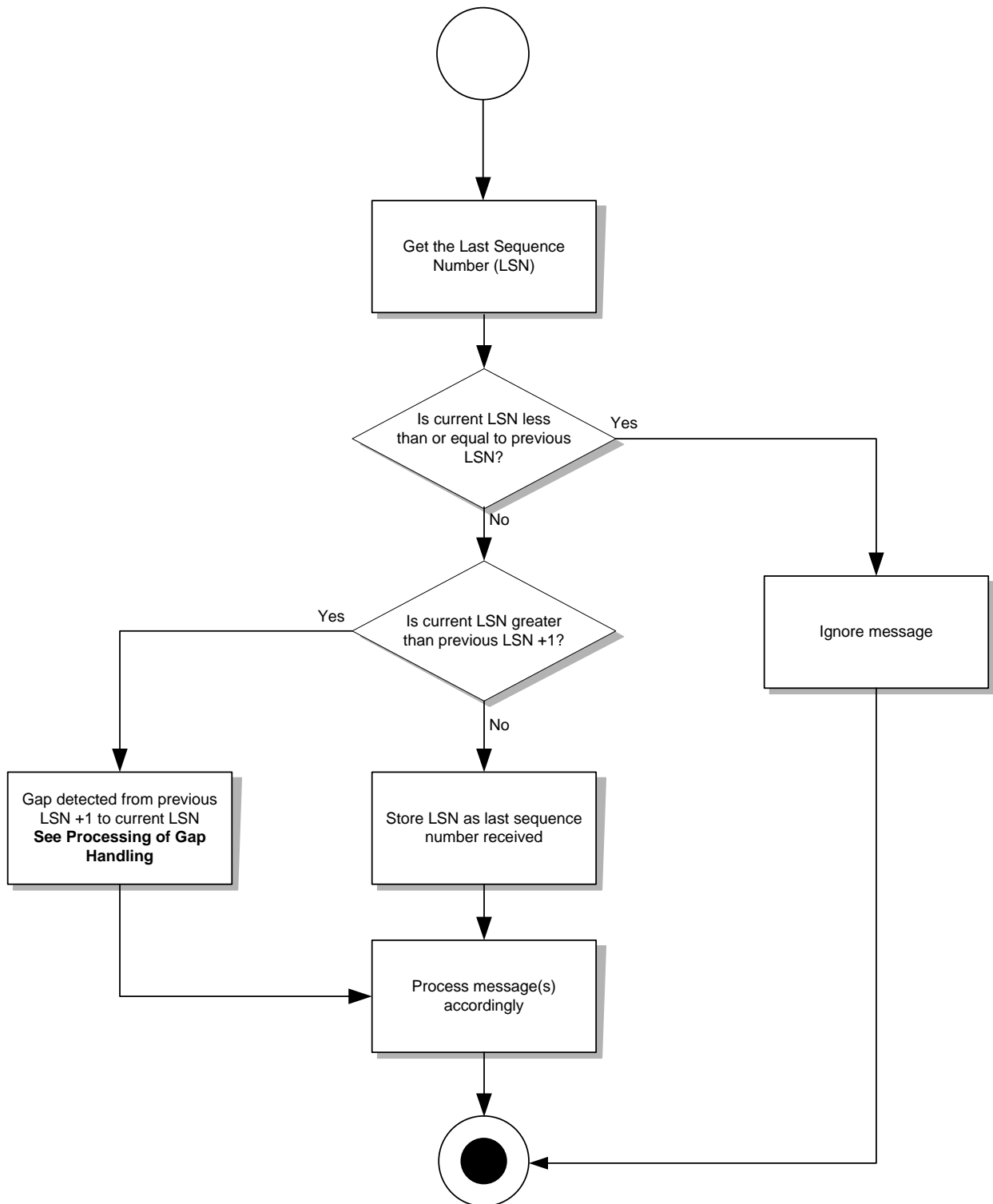
Figure 4 Heartbeat Response Message Processing



B.5. Processing of Data Messages

The following is the recommended way of processing Data messages:

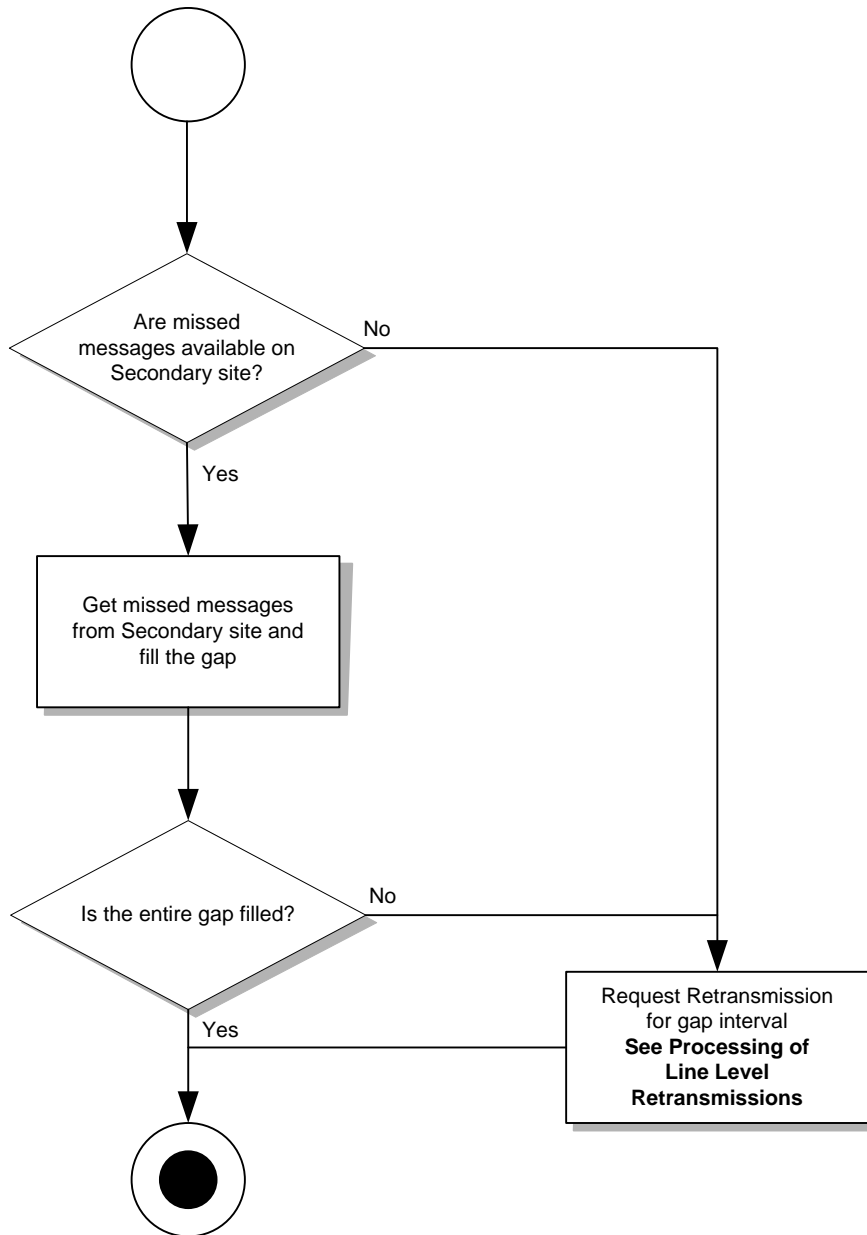
Figure 5 Data Message Processing



B.6. Processing of Gap Handling

The following is the recommended way of handling message gaps:

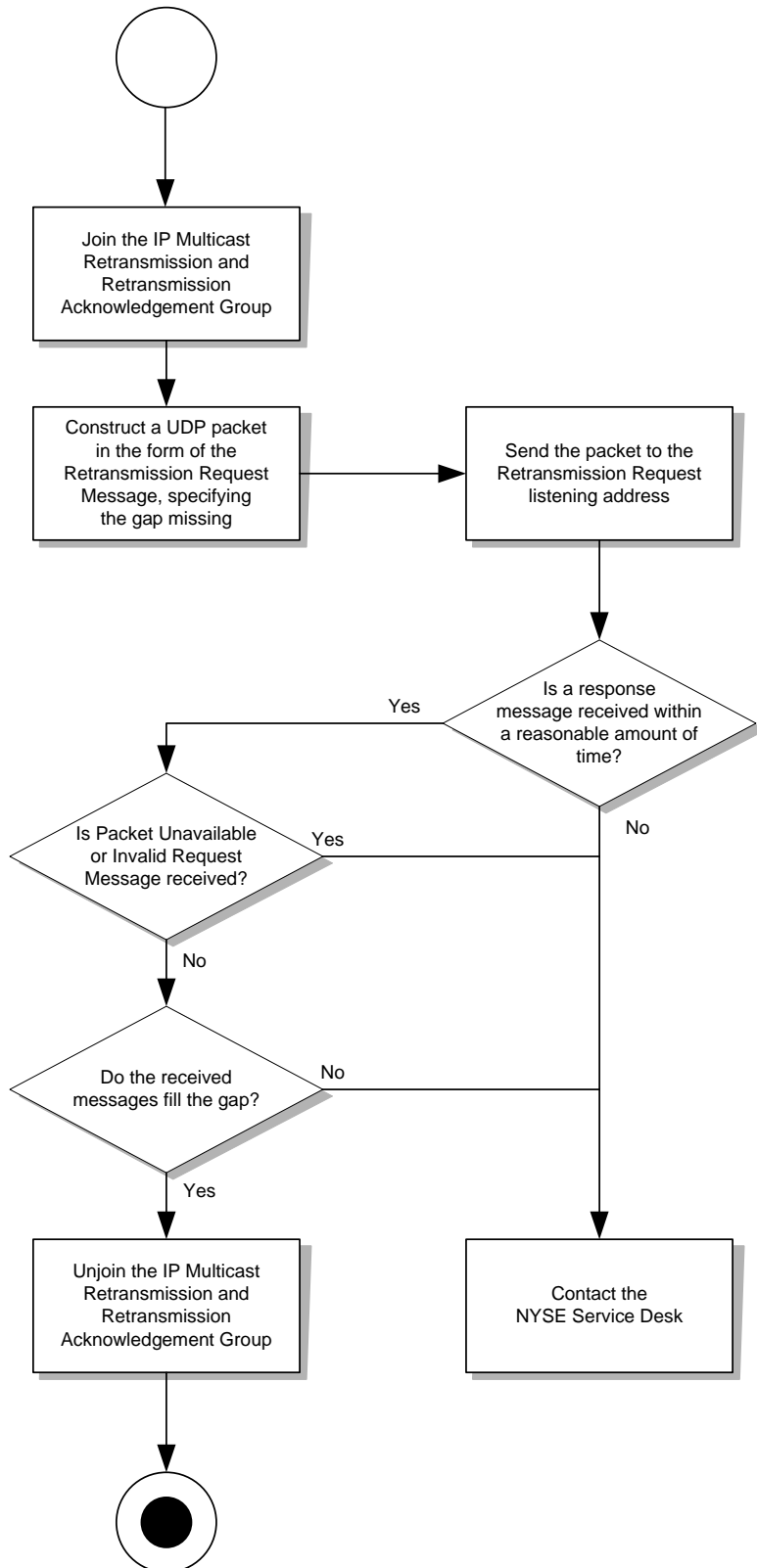
Figure 6 Message Gap Handling



B.7. Processing of Line Level Retransmissions

The following is the recommended way of processing line level retransmissions:

Figure 7 Line Level Retransmission Processing



B.8. Processing of Refresh Messages

The following is the recommended way of processing Refresh messages:

Figure 8 Refresh Message Processing

