Securities Industry Automation Corporation

NYSE OpenBook®
(Real-Time Updates)

Product and Interface Specification

A Member of the PDP Product Suite

VERSION 1.1

Issue Date: 13 July 2004

For more information, please contact:

• NYSE Product:	Ron Jordan	(212) 656-3830
	Mark Schaedel	(212) 656-5418
• NYSE Technical:	Peggy Sullivan	(212) 656-5553
• SIAC Technical:	Valetina Rabinovich	(212) 383-2266
• SIAC Testing:	Nevin Oliphant	(212) 383-3033
	Jim Wieman	(212) 383-7106

DOCUMENT REVISION HISTORY

REVISION	DATE	AUTHOR	DESCRIPTION
0.1	05/07/2004	J. Vigo	Initial Draft.
0.2	05/17/2004	J. Vigo	Initial Review Draft.
0.3	05/19/2004	J. Vigo	PubTime and Timestamp fields' description update.
0.4	07/06/2004	J. Vigo	NYSE inspection changes.
0.5	07/07/2004	J. Vigo	OpenBook 5-Seconds Differences table reformatted.
1.0	07/12/2004	J. Vigo	Product name updated to NYSE OpenBook (Real-Time Updates).
1.1	07/13/2004	J. Vigo	Cover page update.

Table 1. Document Revision History.

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

1. Terms

Table 2, below, defines a number of terms that appear throughout this document.

TERM	DEFINITION/DESCRIPTION
BIG ENDIAN	Refers to which bytes are most significant in multi-byte data types. In big-endian architectures, the leftmost bytes are most significant. This byte order also corresponds to Network Byte Order.
DBK	NYSE Specialist Limit Order Book.
GROUP ID	IP Multicast address for PDP data delivery.
HOST SYMBOL FORMAT	A format set internally by NYSE order processing systems. Its representation of suffix is different from NMS systems.
MART	Message Archive and ReTransmission – The PDP component that archives and retransmits PDP message packets.
MINIMUM PRICE VARIATION (MPV)	Refers to the minimum permissible price movement of a stock symbol. With the inception of decimals, the MPV has generally been a penny.
MONOTONICAL	Incrementally increase in value by one.
PDP	NYSE Proprietary Data Product.
PORT NUMBER	Socket port assigned to a feed.
PUBLISHER	Generic name for any system/application generating PDP message products.
PUMP	Publisher Message Proxy – The data publication engine of the PDP that 'productizes' and publishes PDP message packets.
RECIPIENT	Synonymous with *Subscriber*.
SEQUENCE NUMBER	A unique, sequential message ID that both 'tags' each message and allows recipients to identify message 'gaps' and request retransmission (if appropriate).
SUBSCRIBER	Any customer/client system that will subscribe to, and receive data products from, the PDP data engine (i.e. the PuMP).
SYMBOL STATE	A list of active price points for the stock symbol along with its trading information (transmitted in a Full Update Message).
MTU	Maximum Transfer Unit – The largest size of IP datagram that may be transferred on a given network. Most network implementations have a default setting of 1500 bytes.

TABLE 2. DEFINITION OF TERMS

2. Document Scope

2.1 Identification

This document specifies the methods and procedures by which authorized PDP OpenBook recipients may subscribe to, and receive NYSE OpenBook (Real-Time Updates) data from SIAC.

2.2 Document Overview

This interface specification defines the general session and transport protocol requirements for recipients of PDP OpenBook data.

Many of the general technical concepts referenced herein are detailed in the *Data Distribution Model for IP Multicast Based Environment* specification, which is listed below as reference [1] and in the SFTI document, which is listed below as reference [2].

2.3 Referenced Documents

The following documents are referenced herein:

[1] *Data Distribution Model for IP Multicast Based Environment- Version 1.7*; SIAC Communication Engineering Planning and Development; 9 November 2000.

[2] *http://sfti.siac.com*

2.4 Web Site

For more information regarding the NYSE OpenBook (Real-Time Updates) feed, please reference the following Internet web site:

http://www.nysedata.com/openbook

3. PDP System Overview

3.1 Introduction

The document defines the interface specifications that will identify the procedures by which authorized recipients can interpret PDP data received via or Secure Financial Transaction Infrastructure (SFTI).

Authorized recipients may subscribe to one or more PDP products via any given connection to the SFTI network.

3.2 General Guidelines

The following guidelines will govern the creation and transmission of all NYSE PDP products.

3.2.1 IP ASSIGNMENTS

Note that the following terms are analogous:

- Multicast Group
- Multicast Group ID
- Multicast Host Group
- Multicast Host Group ID

The Multicast groups IDs are represented by a multicast IP address and port number. The multicast IP address is a portion of the IP address range known as Class D IP addresses. Using standard IP address notation, this range covers addresses from 224.0.0.0 through 239.255.255.255.

All PDP services will be assigned a set of unique Multicast Group IDs as described in Table 3. Note that not all services will require (or be provided with) the full suite of address types.

ADDRESS TYPE	DESCRIPTION
Primary Site Multicast Group ID	Multicast Group ID(s) assigned to the product delivered from the primary SIAC site.
Secondary Site Multicast Group ID	Alternate Multicast Group ID(s) assigned to the product delivered from a secondary SIAC site.
Primary Retransmission Multicast Group ID	Multicast Group ID(s) to which retransmitted packets for a specific product are delivered from the primary SIAC site.
Secondary Retransmission Multicast Group ID	Alternate Multicast Group ID(s) to which retransmitted packets for a specific product are delivered from the secondary SIAC site.
Primary Retransmission Requests Unicast Group ID	UDP address to which subscribers can make requests for retransmissions.
Secondary Retransmission Requests Unicast Group ID	Alternate UDP address to which subscribers can make requests for retransmissions.

Table 3. General IP Assignments

All above Addresses are also available for testing during 'after hours' replay of data.

Note that the use of the terms 'primary' and 'secondary' in Table 3 is meant only to reflect dual-site data delivery. Data originating from either site is equally reliable.

3.2.2 GAP DETECTION

The PDP Distribution System will assign all data packets a unique, sequential message ID. This will allow recipients to identify 'gaps' in the message sequence and, if appropriate, reconcile them 'locally' with an alternate feed (Section 3.2.3) or request retransmission of the missing/corrupted data packet (Section 3.2.4).

Note that control messages may not have sequence numbers associated with them. Please refer to the control message section of this specification.

3.2.3 DUAL SITE

Depending on business requirements, individual data products may be published using two (2) sets of unique IP Multicast Group IDs—each originating from a separate SIAC distribution site. Thus, when appropriate, the PuMP will transmit a given message packet over two (2) Multicast Groups, one originating from each site and each containing an identical sequence number. This will allow customers to receive two redundant feeds. Additionally, any message on either feed can be retransmitted upon request (some products may have restrictions, refer to section 3.2.4).

3.2.4 RETRANSMISSION

Some data products may support—upon authorized subscriber request—retransmission of 'missing'[1] data packets. In such cases, subscribers may issue a 'retransmission request'. In response, the MART (a PDP system component) will retransmit the packet as appropriate.[2] Please refer to Section 5. for details on how retransmissions are handled.

3.2.4.1 Retransmission Limitations

Table 4, below, summarizes the Retransmission limitations:

CAPABILITY	DESCRIPTION	THRESHOLD	ACTION
Prevention of invalid subscribers	This feature drops all incoming requests from subscribers that are not in the enabled subscribers list. PDP subscribers will need a source id, which is a string that uniquely identifies the subscriber of the retransmission requests. Please contact NYSE Technical to get a unique source ID.	Product specific.	Request will not be processed.
Limitation of Requests for a large number of packets	Each specific PDP product has a limitation in the number of packets a client can request each time. In case a request exceeds the maximum amount the client can continue to issue requests, but they will be dropped.	Product specific.	Request will not be processed.
Limitation of Refresh Requests	This feature keeps track of the number of subscriber refresh requests. If the number of refresh requests reaches the threshold limit, the subscriber will be added to the blocked subscribers list and its retransmission requests, of any kind, will no longer be accepted.	Product specific.	All subsequent subscribers' retransmissions requests will be blocked.
Limitation of Generic Requests	If the number of subscriber's generic requests reaches the threshold number of requests per day, this subscriber will be blocked and its retransmission requests, of any kind, will no longer be accepted during that particular day.	Product specific.	All subsequent subscribers' retransmissions request will be blocked.

Table 4. Retransmission Limitations

[1] The actual cause of the 'message gap' (e.g., transmission errors, hardware failures, subscriber software error, etc.) is irrelevant. However, SIAC does reserve the right to suspend the retransmission service selectively whenever its use affects 'normal' transmissions.

[2] Details will be provided separately.

NOTE: PDP subscribers should contact SIAC's support if they want to update their source id or are having problems with retransmission requests.

3.2.5 MESSAGE PLAYBACK

The PDP products may offer a message 'replay' service. That is, at a specified time a PDP component (MART) will replay—from a previous session—all messages, or a range of messages, for a given service in their original sequence, and at their original pace rate.

4. Session Protocol

4.1 Session Initiation and Termination

Recipient's applications/hosts will be responsible for issuing Multicast *subscriptions* to one or more of the Multicast Groups assigned to a particular product. In response to each authorized[3] subscription request, SFTI network will complete the tasks associated with delivering the Multicast packets from the NYSE data source to the recipient's network.

The process of subscribing to a Multicast Group ID is also known as 'joining' a Multicast Group. Upon session termination, the subscriber's host system should issue an 'unjoin' message. This will terminate delivery of data to that host's local network. If an application/host terminates without issuing an 'unjoin' message, the network will eventually issue a 'timeout' for the Multicast Group subscription that will automatically terminate delivery of the Multicast packets to the host's local network.

Details regarding the protocols and procedures involved with requesting and receiving Multicast data packets are presented in [1]

4.2 Authentication

Refer to [1] for specific authentication requirements.

4.3 RETRANSMITTED MESSAGES

Retransmitted packets will be identified by the value of 'RETRANSFLAG' field in the message header. Please refer to the Retransmission Messages section below for any additional information.

[3] An authorized subscription is that originated from an authorized client. An authorized client requested the service to NYSE Customer Service and is included in the list of valid subscribers.

5. Common PDP Message Structure

5.1 General Notes

In broad terms, there are two types of messages transmitted as part of this protocol: *control* and *data*. Control messages do not contain data *per se*; rather, they allow conversing parties to exchange session-specific information (e.g., 'reset sequence number'). Data messages are product specific and, although they will adhere to the general specification, they are defined specifically in a later section.

5.2 Format Specifics

All messages are delivered in a wire format as described later in this document.

In addition, the following is true about the delivery format:

- All fields in a message will be contiguous
- No fields are explicitly aligned
- All binary fields will be represented using 'Big Endian' format
- All binary fields will be unsigned (unless otherwise specified)
- A message can span multiple physical packets
- Any physical packet will contain at most one message
- ASCII strings are non-null terminated character arrays (unless otherwise specified)

5.3 Common PDP Message Header

All PDP messages will contain a Common PDP Message Header. 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 5, defines the Common PDP Message Header Format which all PDP products conform to. Note that the fields must appear in the order indicated by their relative position in the table.

FIELD NAME	DESCRIPTION	FORMAT	SIZE (BYTES)
PRODUCTID	Product code/identifier. Uniquely defined for each PDP product. Value of '108' for NYSE OpenBook (Real-Time Updates).	BINARY INTEGER	1

FIELD NAME	DESCRIPTION	FORMAT	SIZE (BYTES)
VERSIONID	Version ID – Used to identify message formats/revisions for a given product.	BINARY INTEGER	1
SEQNUM	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).	BINARY INTEGER	4
MSGTYPE	Message Type Identifier. It's used to identify the type of message so it can be interpreted by the recipient.	BINARY INTEGER	1
RETRANSFLAG	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[4]	BINARY INTEGER	1
TIMESTAMP	This field records/indicates message creation time by PDP. It does not indicate the time of data generation. All retransmissions of this message will retain its original timestamp. The format of this field is as follows: YYYYMMDDhhmmssxxx Where: YYYY – Millennium-compliant year MM – Month (01-12) DD – Day of Month (1-31) hh – Hour of day (00-23) mm – Minute (00-59) ss – Second (00-59) xxx – Milliseconds (000-999)	ASCII STRING	17

[4] Refreshes give current state.

FIELD NAME	DESCRIPTION	FORMAT	SIZE (BYTES)
MSGBODYSIZE	The message body size in bytes.	BINARY INTEGER	2

Table 5. Required PDP Message Header Fields

5.4 Publisher Control Messages

These are control messages that originate from PDP Products. The following control messages are supported as part of this protocol.

5.4.1 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.

5.4.1.1 Message Format

Below, defines the format of the Sequence Number Reset.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Sequence Number Reset Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'1'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'4'	Refer to Table 5.
Defined below are the 'body' fields of the Sequence Number Reset Message				
NEXTSEQNUMBER	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.

Table 6. Sequence Number Reset Message Format

5.4.1.2 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 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). Refer to Figure 2 in Appendix B for a suggested way of processing.

5.4.2 NO DATA AVAILABLE

Whenever possible, NYSE/SIAC systems will transmit a No Data Available message to subscribers to alert them of a processing anomaly.

5.4.2.1 Message Format

Below, defines the format of the No Data Available message.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the No Data Available Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'4'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'0'	Refer to Table 5.

Table 7. No Data Available Message Format

5.4.2.2 Processing Notes

- Subscribers should not expect to receive any No Data Available messages except in case of a processing anomaly.
- No Data Available messages will only contain the PDP Message Header with an empty body.

5.4.3 HEARTBEAT

In the case when data is not being sent, Subscribers will receive heartbeat messages to let them know that the system is still alive.

5.4.3.1 Message Format

Below, defines the format of the Heartbeat message.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Heartbeat Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'2'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'0'	Refer to Table 5.

Table 8. Heartbeat Message Format

5.4.3.2 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.

Refer to Figure 3 in Appendix B for a suggested way of processing.

5.4.4 ADMIN MESSAGE

These are messages used to send administrative messages to the subscribers.

This message is not currently available. It will be enabled in the future.

5.4.4.1 Message Format

Below, defines the format of the Admin message.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Admin Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'3'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER		Refer to Table 5.
Defined below are the 'body' fields of the Admin Message				
TXTMSG	varies	ASCII STRING		Non-null terminated string whose size will be indicated by the MsgBodySize field in the Header.

Table 9. Admin Message Format

5.5 Subscriber Control Messages

These messages will originate from Subscribers. Currently there are no subscriber control messages available.

5.6 Retransmission Messages

The sections below describe the different retransmission related messages.

5.6.1 GENERIC RETRANSMISSION REQUEST MESSAGE

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

5.6.1.1 Message Format

This message will contain the fields/values specified in Table 9, below.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Generic Retransmission Request Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'20'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'28'	Refer to Table 5.
Defined below are the 'body' fields of the Generic Retransmission Request Message				
SOURCEID	20	ASCII STRING		This field represents the name of the source requesting retransmission.
BEGINSEQNUM	4	BINARY INTEGER		The beginning sequence number of the requested range of messages to be retransmitted.
ENDSEQNUM	4	BINARY INTEGER		The end sequence number of the requested range of messages to be retransmitted.

Table 10. Generic Retransmission Request Message Format

5.6.2 INVALID GENERIC REQUEST MESSAGE

This message will be sent to inform the subscribers of an invalid request for a range of messages that they requested retransmission for.

This message is not currently available. It will be enabled in the future.

5.6.2.1 Message Format

This message will contain the fields/values specified in Table 10, above.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Invalid Generic Request Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
MSGTYPE	1	BINARY INTEGER	'31'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'28'	Refer to Table 5.
Defined below are the 'body' fields of the Invalid Generic Request Message				
REASONINT	2	BINARY INTEGER		Predefined reason represented in an integer. 1 – Begin sequence number is larger than end sequence number. 2 – Request range exceeds number of allowed messages in a single request.
REASONSTR	18	ASCII STRING		Reason represented in string format.
BEGINSEQNUM	4	BINARY INTEGER		The beginning sequence number of the requested range of messages to be retransmitted.
ENDSEQNUM	4	BINARY INTEGER		The end sequence number of the requested range of messages to be retransmitted.

Table 11. Invalid Generic Request Message Format

5.6.3 RETRANSMITTED 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 PuMP, with the exception that the 'RetransFlag' in the header is set to the value of '2' or '4' depending on whether the retransmission is for a non-replay or a replay retransmission message, respectively.

5.6.3.1 Message Format

Table 12, below, defines the format of the Retransmitted message.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the Retransmitted Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY IN-		Refer to Table 5.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
		TEGER		
MSGTYPE	1	BINARY INTEGER		It will be the MsgType of the original message sent by the PuMP.
RETRANSFLAG	1	BINARY INTEGER	'2' or '4'	Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER		Refer to Table 5.
All the 'body' fields of the Retransmitted Message are the same as the original message				

Table 12. Retransmitted Message Format

5.6.3.2 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.

Refer to Figure 6 in Appendix B for a suggested way of processing.

5.6.4 MESSAGE UNAVAILABLE

This message will be sent to inform the subscribers of unavailability of a range of messages that they requested retransmission.

5.6.4.1 Message Format

Below, defines the Message Unavailable message.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the NYSE Packet Unavailable Message				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'5'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.

FIELD NAME	SIZE	FORMAT	VALUE	DESCRIPTION
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	'8'	Refer to Table 5.
Defined below are the 'body' fields of the NYSE Packet Unavailable Message				
BEGINSEQNUM	4	BINARY INTEGER		The beginning sequence number of the requested range of messages to be retransmitted.
ENDSEQNUM	4	BINARY INTEGER		The end sequence number of the requested range of messages to be retransmitted.

Table 13. Message Unavailable Message Format

6. NYSE OpenBook (Real-Time Updates) Message Specifications

6.1 Overview

The NYSE OpenBook (Real-Time Updates) service provides Subscribers all price points for NYSE Listed Securities. The data found in this service is generated by the DBK system and provides a means to construct the "look at the book" accurately for any stock symbol (generated with up to 1 second granularity). The NYSE OpenBook (Real-Time Updates) product provides a view of the Specialist's limit order book for each DBK stock.

6.2 Delivery Format

The NYSE OpenBook (Real-Time Updates) service uses the poll-based publishing model. This means that data will be published based on a specified publication frequency (up to 1 second), i.e., the data will be current at the time of publication.

The NYSE OpenBook (Real-Time Updates) service sends the complete state (all price points) for each NYSE listed security in the form of a Full Update Message (one per security) at the beginning of the session. From that point onward it sends the change in state (price points that changed) in the form of Delta Update Messages at a pre-defined frequency. In case of an error a new session is started, which can be detected by a change in the Internal Sequence Number (refer to 6.4.3 for details).

The following is a description of the types of messages:

Message Type	Description
Full Update Message	Contains all the price points for a particular stock symbol.
Delta Update Message	Contains price point changes that have occurred (e.g., insertion, modification or deletion) for a particular stock symbol since the last Full Update or Delta Update Message.

Table 14. NYSE OpenBook (Real-Time Updates) Message Types

For more information about the message format, processing information, and/or contents of the message, please refer to section 6.8.

6.3 Data Messages

As its name implies, a Full Update Message contains all active price points for a given stock symbol. In the context of this specification, an 'active price point' is one that has at least one open order associated with it at the time the message is published. Thus, if a given price point does not appear in a Full Update Message, there are no orders at that price.

In contrast, a Delta Update Message contains only price points that differ from the most recently published Full Update or Delta Update Message for a given stock symbol. That is, a Delta Update Message contains only the changes that have occurred from the previously published message for

that security. Thus, to remain consistent with the values contained in DBK, Subscribers must apply all Delta Update Messages in publication sequence based on symbol sequence number.

Note: If during the prior publication interval a given price point no longer contains at least one (1) open order[5], the Delta Update Message will publish a zero (0) quantity value for that price point.

In addition, it is recommended that a Subscriber replace its previous stock symbol state with the data in a Full Update Message whenever it is received.

6.4 Sequence Numbers

All messages conform to the line level sequencing described in Table 5. Subscribers can use sequence numbers to determine the following:

- Missing (gapped) messages
- Misordered messages
- Duplicate messages

6.4.1 PACKET SEQUENCE NUMBERS

The number of active price points (particularly that of Full Updates) is unbounded, so NYSE OpenBook (Real-Time Updates) data messages may span multiple packets. The format for NYSE OpenBook (Real-Time Updates) packet sequence numbers is set forth below.

- Two (2) unsigned one-byte binary integers, where:
  - The first unsigned binary integer represents the packet sequence number (for multi-packet messages).
  - The second unsigned binary integer indicates the total number of packets in this logical message (for multi-packet messages).

6.4.1.1 Packet Sequence Number Example

Table 15, below, provides an example of a NYSE OpenBook (Real-Time Updates) data message that requires three (3), one (1), two (2), and (1) message packets.

Line Sequence Number	Packet Sequence Number (1 Byte)	Number of Packets (1 Byte)	Symbol	Corresponding Symbol Sequence Number (SSN)
1	1	3	ABC	1
2	2	3	ABC	1
3	3	3	ABC	1
4	1	1	DEF	1
5	1	2	DEF	2
6	2	2	DEF	2
7	1	1	ABC	2

[5] For example, all orders for that price point were executed or cancelled during the preceding interval.

Table 15. Example Packet Sequence Numbering.

Note: Symbols used in this Table are not actual symbols, and are just used for example purposes.

6.4.2 SYMBOL-LEVEL SEQUENCE NUMBERS

Since Full Update and Delta Update Messages are applied per stock symbol, this service provides for unique sequence numbers per stock symbol. These sequence numbers are in addition to the line-level sequence numbers defined in the header of each packet. If a line-level sequence gap is detected, the symbol based sequence numbers can be used to determine which symbols were affected.

The format for NYSE OpenBook (Real-Time Updates) symbol-level sequence numbers consists of two parts and is set forth below.

- The first is four-byte unsigned binary integer (in *Big Endian* format) which represents the Symbol Sequence Number (SSN). This number will increase monotonically with every message published for a particular stock symbol.
- The second is one-byte unsigned binary integer. This field will represent a unique value (SSN session) that can be used to determine if the current SSN values are the same as previous SSN values. Given the fact that an SSN can reset during the day due to failures, recoveries, or maintenance, it is hard to determine if the SSN relates to a previous SSN (when processing gaps). Using the SSN session Id, if the value in this field is equal to a previously received message value in the same field, the messages are part of the same symbol sequence chain.

In the event of failures or internal operations procedures (e.g., in the event of a DBK restart), the SSN can be reset to the value of one (1) for any stock symbol. As a result, the first message with SSN of 1 (one) will be a Full Update Message for that stock.

6.4.3 SESSION SEQUENCE NUMBERS

NYSE OpenBook (Real-Time Updates) has only one session sequence number, the Internal Sequence Number (ISN). Refer to Table 17 for format information.

The ISN can be used to detect message loss in conjunction with the symbol sequence numbers (Refer to 6.4.2) as shown in Figure 7.

The ISN contains the startup time of the NYSE OpenBook (Real-Time Updates) service and every time the service is restarted, a new session id will be set to ISN following the algorithm below:

ISN = (Start Hour * 10) + (Start Minute / 6)

As an example, if NYSE OpenBook (Real-Time Updates) is started at 7:30 am, ISN will have its value set to 75.

6.5 Symbols

The stock symbols represented in this feed use Host Symbol Format, which uses a 16-bit ASCII character (blank padded) to represent a symbol.

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^^^^^^^^^", where '^' represents a blank space. If the symbol is "ABC" without a suffix, it will be represented as: "ABC^^^^^^^^^^^^". Note that there is no static position for the suffix.

6.6 Price Format

The Price Format represents all prices in a Full Update or Delta Update Message. Any price can be derived from Price Format by using the Price Numerator value, and Denominator.

The formula for conversion is as follows:

Price (Whole) = (Whole part) (PriceNumer / Denominator)

Price (Decimal) = (Whole part) (PriceNumer mod Denominator)

Table 16. Example Price Format., below, provides examples of the Price Format.

PriceNumer	Denom	Price (Whole)	Price (Decimal)	Price (Whole & Decimal)
1025	100	10	25	$10.25
410	100	4	10	$4.10

Table 16. Example Price Format.

6.7 Price Points

Display Book contains a collection of price points on the buy side and the sell side. A price point consists of the following pieces of data (also known as the price point triplet):

- Price (in each message, represented in numerator of the price)
- Quantity (in each message, represented in round lots for the given Price)
- ***Number of Orders* (number of orders for the given Price. This field is reserved for future use, at the present time this value will be zero)**

A Full Update Message will contain the complete list of active price points for both the buy and sell side.

Note: A Delta Update Message will contain the revised values for Quantity for a particular price. For a price that is no longer active, a Delta Update Message will contain a triplet at that price with the Quantity equal to zero (0). If there are no changes to the price points of a stock at a particular delta update publication, no message will be published.

6.8 NYSE OpenBook (Real-Time Updates) Update Data Messages

As discussed previously in this document, the NYSE OpenBook (Real-Time Updates) data is primarily distributed using Full Update and Delta Update Messages. This section describes the following:

- Message Format
- General Processing
- Full Update Specific Processing
- Delta Update Specific Processing

6.8.1 UPDATE MESSAGE FORMAT

The message body contains NYSE OpenBook (Real-Time Updates) data published in a proprietary, binary format. Table 17, below, defines its format and content.

Note: As described in Section 6.4.1, this message may span multiple packets.

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the NYSE OpenBook (Real-Time Updates) Full Update Message.[6]				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'100' '101'	Header field: The MSGTYPE value for a Full Update Message is '100'. The MSGTYPE value for a Delta Update Message is '101'.
RETRANSFLAG	1	BINARY INTEGER		Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER		Refer to Table 5.
Defined below are 'body' fields of the NYSE OpenBook (Real-Time Updates) Full Update Message.				
SSN	4	BINARY INTEGER		Symbol Sequence Number. Refer to Section 6.4.2 for processing details.
ISN	1	BINARY INTEGER		Internal Sequence Number. Refer to Section 6.4.2 for processing details.

[6] Refer to Table 5 for a complete definition of the PDP Common Message Header.

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
PSN	1	BINARY INTEGER		Packet Sequence Number. Refer to Section 6.4.1 for processing details.
NPS	1	BINARY INTEGER		Number of packets in sequence. Refer to Section 6.4.1 for processing details.
PUBTIME	6	ASCII STRING	000000 - 235959	Publisher Time specifies the time the message is published. It corresponds to data generation time. Its format is defined as HHMMSS, where: HH – Hour (00-23) MM – Minute (00-59) SS – Second (00-59)
SYMBOL	16	ASCII STRING	Any valid NYSE symbol	Symbol. Refer to Section 6.5 for processing details.
TRADINGINDICATOR	1	ASCII STRING	'T' or 'H'	Trading Indicator. 'T' – stock symbol is trading. 'H' – stock symbol is halted.
MPV	2	BINARY INTEGER		Minimum Price Variation.
UOT	2	BINARY INTEGER		Unit of Trade. This value represents the general unit of trade at which the stock symbol trades.
PDENOM	2	BINARY INTEGER		Price Denominator. Refer to section 6.6 for details.
LASTSALEPRICE	4	BINARY INTEGER		Last Sale Price Numerator. Represents the last price of the sale.
NUMBUYPOINTS	2	BINARY INTEGER		Number of Buy Points in this packet.
NUMSELLPOINTS	2	BINARY INTEGER		Number of Sell Points in this packet.
BUYPRICENUMER	4	BINARY INTEGER		Buy Price Numerator. Represented using the Price Format. Refer to Section 6.6 for processing details.
BUYQUANTITY	2	BINARY INTEGER		Buy Quantity. This value represents the quantity (in terms of UOT) there are for this price point.
BUYNUMORDERS	2	BINARY INTEGER		Buy Number of Orders. Reserved for future use, currently set to zero.
SELLPRICENUMER	4	BINARY INTEGER		Sell Price Numerator. Refer to Section 6.6 for processing details.
SELLQUANTITY	2	BINARY IN-		Sell Quantity. This value represents the quan-

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
		TEGER		tity (in terms of UOT) there are for this price point.
SELLNUMORDERS	2	BINARY INTEGER		Sell Number of Orders. Reserved for future use, currently set to zero.

Table 17. Update Message Format.

6.8.2 GENERAL PROCESSING NOTES

The following processing notes apply to both the Full Update and Delta Update Message. Please refer to section 6.8.3 and 6.8.4 accordingly, for additional information. Refer to Appendix B, Figure 7 for a suggested way on how to process Update Messages.

- Any Update Message can span multiple 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 packets, all fields not part of the 'price point triplet' section will be repeated for each packet. To determine the number of 'price point triplets' in any given packet, use the value in the NumBuyPoints and NumSellPoints fields.
- Only field values will appear in the published messages (e.g. no names, 'tags', sizes will appear in the message). The field tags that appear in Table 17, are for reference purposes only.
- In the same vein as the prior bullet, all fields are contiguous. That is, there is no explicit (or implicit) 'padding' between fields regardless of the juxtaposed data types, sizes, and alignment issues.
- If, for a given entry, the Value Field in Table 17, above, is blank, it means that the value for that field is not 'fixed', but rather are computed and thus will vary with each message.
- All field sizes are fixed and constant.
- The publication time referenced in Table 17, above, will be using Eastern Standard Time (EST).
- Binary fields are provided in *Big Endian* format.
- ASCII string fields are not null terminated.

6.8.3 FULL UPDATE PROCESSING NOTES

The following notes provide general guidelines for processing NYSE OpenBook (Real-Time Updates) Full Update Messages. The reader should not ascribe any importance to the order in which they are presented.

- Full Update Messages that span multiple packets 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.

6.8.4 DELTA UPDATE PROCESSING NOTES

The following notes provide general guidelines for processing NYSE OpenBook (Real-Time Updates) Delta Update Messages. The reader should not ascribe any importance to the order in which they are presented.

- 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.

6.9 Retransmission and Refresh Support

The NYSE OpenBook (Real-Time Updates) service provides a mechanism to retransmit line level sequence numbers and Full Update refresh messages. This section explains how to interact with this mechanism and its limitations.

6.9.1 RETRANSMISSION LIMITATIONS

Table 18,below, summarizes the Retransmissions Limitations for NYSE OpenBook (Real-Time Updates).

CAPABILITY	DESCRIPTION	THRESHOLD	ACTION
Prevention of invalid subscribers	See Table 4.	N/A	See Table 4.
Limitation of Requests for a large number of packets	See Table 4.	1000 per request	See Table 4.
Limitation of Refresh Requests	See Table 4.	5000 per day	See Table 4.
Limitation of Generic Requests	See Table 4.	500 per day	See Table 4.

Table 18. NYSE OpenBook (Real-Time Updates) Retransmission Limitations

6.9.2 RETRANSMISSION MESSAGES

All the retransmission messages have the PDP header followed by the body. The types of retransmission messages are as follows:

- OpenBook Refresh Request Message: This message enables Subscribers to request for the current state of a symbol. The response to this message will be a refresh message for the symbol.
- OpenBook Refresh Message: This is a snapshot of the current state of the symbol.

6.9.3 OPENBOOK REFRESH REQUEST MESSAGE

The OpenBook refresh request message has the following fields:

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the NYSE OpenBook (Real-Time Updates) Refresh Request Message[7]				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'22'	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER	'1'	Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	36	Refer to Table 5.
Defined below are 'body' fields of the NYSE OpenBook (Real-Time Updates) Refresh Request Message.				
SOURCEID	20	ASCII STRING		The name of the requestor (used by SIAC to identify who is requesting retransmissions).
SYMBOL	16	ASCII STRING		Any valid NYSE symbol.

Table 19. OpenBook Refresh Request Message

[7] Refer to Table 5 for a complete definition of the PDP Common Message Header.

6.9.4 OPENBOOK INVALID REFRESH REQUEST MESSAGE

This message will be sent to inform the subscribers of an invalid request for a symbol that they requested retransmission.

The OpenBook invalid refresh request message has the following fields:

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the NYSE OpenBook (Real-Time Updates) Invalid Refresh Request Message [8]				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER		Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'31`	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER	'1'	Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER	36	Refer to Table 5.
Defined below are 'body' fields of the NYSE OpenBook (Real-Time Updates) Invalid Refresh Request Message.				
REASONINT	2	BINARY INTEGER		Predefined reason represented in an integer. 1 – No information available. 2 – Invalid symbol.
REASONSTR	18	BINARY INTEGER		Reason represented in a string.
SYMBOL	16	ASCII STRING		Symbol that was requested.

Table 20. OpenBook Invalid Refresh Request Message

6.9.5 OPENBOOK REFRESH MESSAGE

The OpenBook Refresh message is a snapshot of the current state of the symbol. The Refresh message has the same format as the Full Update Message; they only differ in the value of the RetransFlag field. The symbol-level sequence number (SSN) in this message indicates the

[8] Refer to Table 5 for a complete definition of the PDP Common Message Header.

symbol-level sequence number of the last message that was applied to generate the refresh message. Refer to Figure 8 in Appendix B for a suggested way of processing.

Note: Due to the separation of the data feed and the retransmission feed, live messages may be received before the refresh message is received. The live messages must be saved until the refresh message is received. The symbol-level sequence number in the refresh message will indicate which messages were applied to it, and any live messages with symbol-level sequence number lower than the symbol-level sequence number of the refresh message can be disregarded, and the live messages with later numbers must be applied to the refresh.

FIELD NAME	SIZE (BYTES)	FORMAT	VALUE	DESCRIPTION
Set forth below are the 'header' fields of the NYSE OpenBook (Real-Time Updates) Refresh Message [9]				
PRODUCTID	1	BINARY INTEGER	'108'	Refer to Table 5.
VERSIONID	1	BINARY INTEGER	'2'	Refer to Table 5.
SEQNUM	4	BINARY INTEGER	0	Refer to Table 5.
MSGTYPE	1	BINARY INTEGER	'100`	Refer to Table 5.
RETRANSFLAG	1	BINARY INTEGER	'5'	Refer to Table 5.
TIMESTAMP	17	ASCII STRING		Refer to Table 5.
MSGBODYSIZE	2	BINARY INTEGER		Refer to Table 5.
All the 'body' fields of the NYSE OpenBook (Real-Time Updates) Refresh Message are the same as the Full Update Message				

Table 21. OpenBook Refresh Message

[9] Refer to Table 5 for a complete definition of the PDP Common Message Header.

7. IP Group Assignments

7.1 Production Settings

Production IP/Multicast Group Assignments

Table 22, below, defines the IP/Multicast Group and Port assignments for all messages in this service.

ADDRESS TYPE	GROUP IP	PORT NO.	DESCRIPTION
Primary OpenBook Feed	224.0.5212	8212	Primary data feed originating from SIAC.
Secondary OpenBook Feed	224.0.5.213	8213	Secondary data feed originating from SIAC.
Primary Retransmission Service	224.0.5.214	8214	Primary retransmission data feed originating from SIAC.
Secondary Retransmission Service	224.0.5.215	8215	Secondary retransmission data feed originating from SIAC.

Table 22. NYSE OpenBook (Real-Time Updates) Production IP Group Assignments

7.1.1 PRODUCTION RETRANSMISSION REQUEST IP ASSIGNMENTS

Table 23, below, defines the UDP IP Address and Port assignments for the retransmission request service.

ADDRESS TYPE	GROUP IP	PORT NO.	DESCRIPTION
Primary Retransmission Request Address	198.140.59.88	8800	Primary retransmission request address.
Secondary Retransmission Request Address	198.140.58.88	8800	Secondary retransmission request address.

Table 23. NYSE OpenBook (Real-Time Updates) Production Retransmission Request IP/Port

7.2 Replay/Test Settings

This section contains the IP/Multicast Group Assignments and Retransmission Request Assignments for the test configuration. These tests are generally run at night and out of band from the production environment so that Subscribers need not worry about incorrect data on the production lines.

Replay/Test IP/Multicast Group Assignments

Table 24, below, defines the IP/Multicast Group and Port assignments for all messages in this service.

ADDRESS TYPE	GROUP IP	PORT NO.	DESCRIPTION
Primary OpenBook	224.0.5.216	8216	Primary data feed originating

ADDRESS TYPE	GROUP IP	PORT NO.	DESCRIPTION
Replay Feed			from SIAC.
Secondary OpenBook Replay Feed	224.0.5.217	8217	Secondary data feed originating from SIAC.
Primary Retransmission Service Replay Feed	224.0.5.218	8218	Primary retransmission test data feed originating from SIAC.
Secondary Retransmission Service Replay Feed	224.0.5.219	8219	Secondary retransmission test data feed originating from SIAC.

Table 24. NYSE OpenBook (Real-Time Updates) Test IP Group Assignments

7.2.1 REPLAY/TEST RETRANSMISSION REQUEST IP ASSIGNMENTS

Table 25, below, defines the UDP IP Address and Port assignments for the retransmission request service.

ADDRESS TYPE	GROUP IP	PORT NO.	DESCRIPTION
Primary Replay Retransmission Request Address	198.140.59.88	8900	Primary retransmission test request address.
Secondary Replay Retransmission Request Address	198.140.58.88	8900	Secondary retransmission test request address.

Table 25. NYSE OpenBook (Real-Time Updates) Test Retransmission Request IP/Port

8. Publication Frequencies

The Publisher will generate the messages specified herein as per the frequencies specified in Table 26, below.

MESSAGE TYPE	PUBLICATION FREQUENCY
Full Update	The publication of the first full set of Full Update Messages will at 7:30 AM. This full set will contain the complete set of NYSE symbols. After the start up, Full Update Messages will be sent only upon a failure or due to an operational procedure.
Delta Update	A delta message for each stock will begin 1-second after the publication of the full update for that stock and will repeat every 1-second. The exception to the 1-second publication interval is when no activity for a stock occurs. In this case, the publication interval is skipped. If activity occurs after that, the next 1-second publication will again contain the Delta Update. The publication of delta update messages will end 20 minutes after the Market close, typically at 4:20 PM.

Table 26. NYSE OpenBook (Real-Time Updates) Publication Frequency

APPENDIX A

This section provides examples of the content of the data that will be received for Full Update and Delta Update Messages.

The following examples provide a set of mock quotes to demonstrate the NYSE OpenBook (Real-Time Updates) feed based on version two (2). The Symbol used for these scenarios is ABC.

Example Scenarios:

SCENARIO	MESSAGE TYPE	DESCRIPTION
1	Full Update	Send Full Update Message
2	Delta Update	Revise Quantity
3	Delta Update	Remove three price points (Buy side), add one price point (Sell side)
4	No Message	No change in state of stock
5	Delta Update	Remove all price points
6	Delta Update	Add two price points (Buy and Sell side)
7	Delta Update	Add four price points and remove one price point (Buy side), and add two price points (Sell side). Price points split over 3 packets, for example purposes only.
8	Delta Update	Add one price point (Sell side). Price points split over 2 packets, for example purposes only.

Scenario 1: (Full Update)

ABC				
	BUY	46.40	2,000	0
		47.66	1,000	0
		48.20	4,000	0
		49.00	5,000	0
	SELL	49.50	7,000	0
		50.00	2,000	0
		51.00	4,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	1
MSGTYPE	100
RETRANSFLAG	1
TIMESTAMP	20020109073000000
MSGBODYSIZE	100
SSN	1
ISN	0
PSN	1
NPS	1
PUBTIME	073000
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	4
NUMSELLPOINTS	3
BUYPRICE	4640
BUYQUANTITY	20
BUYNUMORDERS	0
BUYPRICE	4766
BUYQUANTITY	10
BUYNUMORDERS	0
BUYPRICE	4820
BUYQUANTITY	40
BUYNUMORDERS	0
BUYPRICE	4900
BUYQUANTITY	50
BUYNUMORDERS	0
SELLPRICE	4950
SELLQUANTITY	70
SELLNUMORDERS	0
SELLPRICE	5000
SELLQUANTITY	20
SELLNUMORDERS	0
SELLPRICE	5100
SELLQUANTITY	40

SELLNUMORDERS	0

Scenario 2: (Delta Update – Revised Quantity)

ABC				
	BUY	46.40	2,000	0
		47.66	1,000	0
		48.20	**6,000**	0
		49.00	5,000	0
	SELL	49.50	7,000	0
		50.00	2,000	0
		51.00	4,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	2
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073010000
MSGBODYSIZE	52
SSN	2
ISN	0
PSN	1
NPS	1
PUBTIME	073010
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	1
NUMSELLPOINTS	0
BUYPRICE	4820
BUYQUANTITY	60
BUYNUMORDERS	0

Scenario 3: (Delta Update – Remove 3 Buy-side price points, add 1 Sell-side price point)

ABC				
	BUY	47.66	1,000	0
	SELL	48.21	5,000	0
		49.50	7,000	0
		50.00	2,000	0
		51.00	4,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	3
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073030000
MSGBODYSIZE	76
SSN	3
ISN	0
PSN	1
NPS	1
PUBTIME	073030
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	3
NUMSELLPOINTS	1
BUYPRICE	4640
BUYQUANTITY	0
BUYNUMORDERS	0
BUYPRICE	4820
BUYQUANTITY	0
BUYNUMORDERS	0
BUYPRICE	4900
BUYQUANTITY	0
BUYNUMORDERS	0
SELLPRICE	4821
SELLQUANTITY	50
SELLNUMORDERS	0

Scenario 4: (No Message – No change in state of stock)

ABC				
	BUY	47.66	1,000	0
	SELL	48.21	5,000	0
		49.50	7,000	0
		50.00	2,000	0
		51.00	4,000	0

Note: No Message is generated for Scenario 4.

Scenario 5: (Delta Update – Remove all price points)

ABC				
	BUY			
	SELL			

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	4
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073050000
MSGBODYSIZE	84
SSN	4
ISN	0
PSN	1
NPS	1
PUBTIME	073050
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	1
NUMSELLPOINTS	4
BUYPRICE	4766
BUYQUANTITY	0
BUYNUMORDERS	0
SELLPRICE	4821
SELLQUANTITY	0
SELLNUMORDERS	0
SELLPRICE	4950
SELLQUANTITY	0
SELLNUMORDERS	0
SELLPRICE	5000
SELLQUANTITY	0
SELLNUMORDERS	0
SELLPRICE	5100
SELLQUANTITY	0
SELLNUMORDERS	0

Scenario 6: (Delta Update – Add two price points on both sides)

ABC				
	BUY	31.44	1,000	0
		31.79	3,000	0
	SELL	32.47	3,000	0
		33.15	4,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	5
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073100000
MSGBODYSIZE	76
SSN	5
ISN	0
PSN	1
NPS	1
PUBTIME	073100
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	2
NUMSELLPOINTS	2
BUYPRICE	3144
BUYQUANTITY	10
BUYNUMORDERS	0
BUYPRICE	3179
BUYQUANTITY	30
BUYNUMORDERS	0
SELLPRICE	3247
SELLQUANTITY	30
SELLNUMORDERS	0
SELLPRICE	3315
SELLQUANTITY	40
SELLNUMORDERS	0

Scenario 7: (Delta Update – Add 4 Buy-side, remove one Buy-side, and add 2 Sell-side price points over 3 packets)

ABC				
	BUY	31.44	1,000	0
		32.82	6,000	0
		32.86	4,000	0
		33.25	1,000	0
		33.61	8,000	0
	SELL	32.47	3,000	0
		33.15	4,000	0
		34.27	1,000	0
		34.65	6,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	6
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073110000
MSGBODYSIZE	68
SSN	6
ISN	0
PSN	1
NPS	3
PUBTIME	073110
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	3
NUMSELLPOINTS	0
BUYPRICE	3179
BUYQUANTITY	0
BUYNUMORDERS	0
BUYPRICE	3282
BUYQUANTITY	60
BUYNUMORDERS	0
BUYPRICE	3286
BUYQUANTITY	40
BUYNUMORDERS	0

Scenario 7 (Cont'd)

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	7
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073110000
MSGBODYSIZE	68
SSN	6
ISN	0
PSN	2
NPS	3
PUBTIME	073110
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	2
NUMSELLPOINTS	1
BUYPRICE	3325
BUYQUANTITY	10
BUYNUMORDERS	0
BUYPRICE	3361
BUYQUANTITY	80
BUYNUMORDERS	0
SELLPRICE	3427
SELLQUANTITY	10
SELLNUMORDERS	0

Scenario 7 (Cont'd)

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	8
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109073110000
MSGBODYSIZE	52
SSN	6
ISN	0
PSN	3
NPS	3
PUBTIME	073110
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	0
NUMSELLPOINTS	1
SELLPRICE	3465
SELLQUANTITY	60
SELLNUMORDERS	0

Scenario 8: (Delta Update – Add one Sell-side price point)

ABC				
	BUY	31.44	1,000	0
		32.82	6,000	0
		32.86	4,000	0
		33.25	1,000	0
		33.61	8,000	0
	SELL	32.47	3,000	0
		33.15	4,000	0
		34.27	1,000	0
		34.65	6,000	0
		35.17	12,000	0

FIELD NAME	VALUE
PRODUCTID	101
VERSIONID	2
SEQNUM	9
MSGTYPE	101
RETRANSFLAG	1
TIMESTAMP	20020109093120000
MSGBODYSIZE	52
SSN	7
ISN	0
PSN	1
NPS	1
PUBTIME	073120
SYMBOL	ABC
TRADINGINDICATOR	T
MPV	1
UOT	100
PDENOM	100
LASTSALEPRICE	4520
NUMBUYPOINTS	0
NUMSELLPOINTS	1
SELLPRICE	3517
SELLQUANTITY	120
SELLNUMORDERS	0

APPENDIX B

Processing of Messages

The following is the recommended way of processing messages:

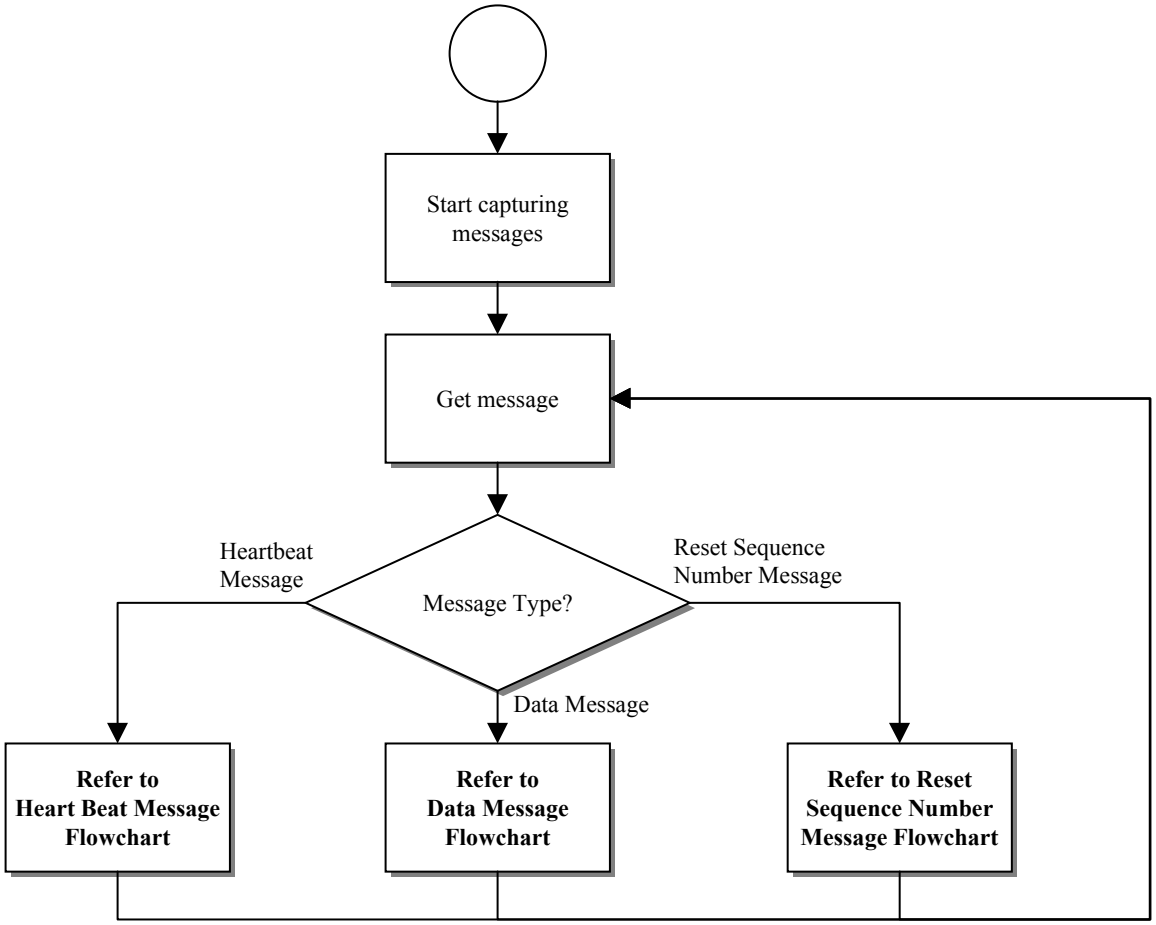

Figure 1. Processing of Messages

Processing of Sequence Number Reset Message

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

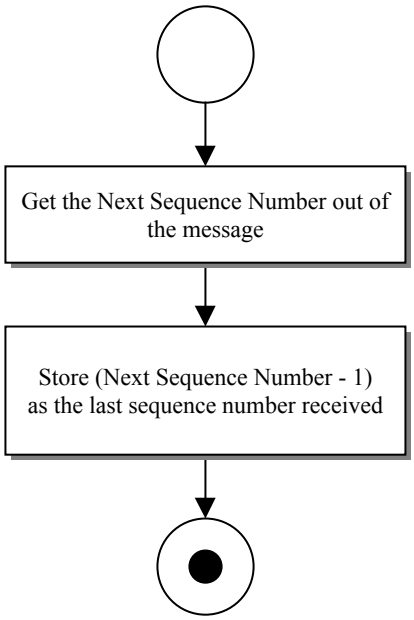

Figure 2. Processing of Sequence Number Reset Message

Processing of Heartbeat Messages

The following is the recommended way of processing Heartbeat Messages:

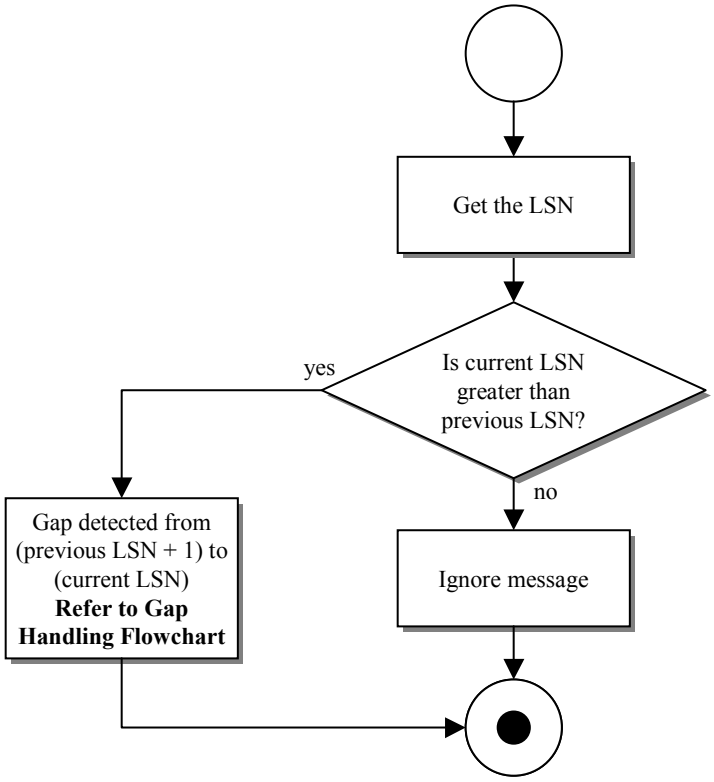

Figure 3. Processing of Heartbeat Messages

Processing of Data Messages

The following is the recommended way of processing data messages:

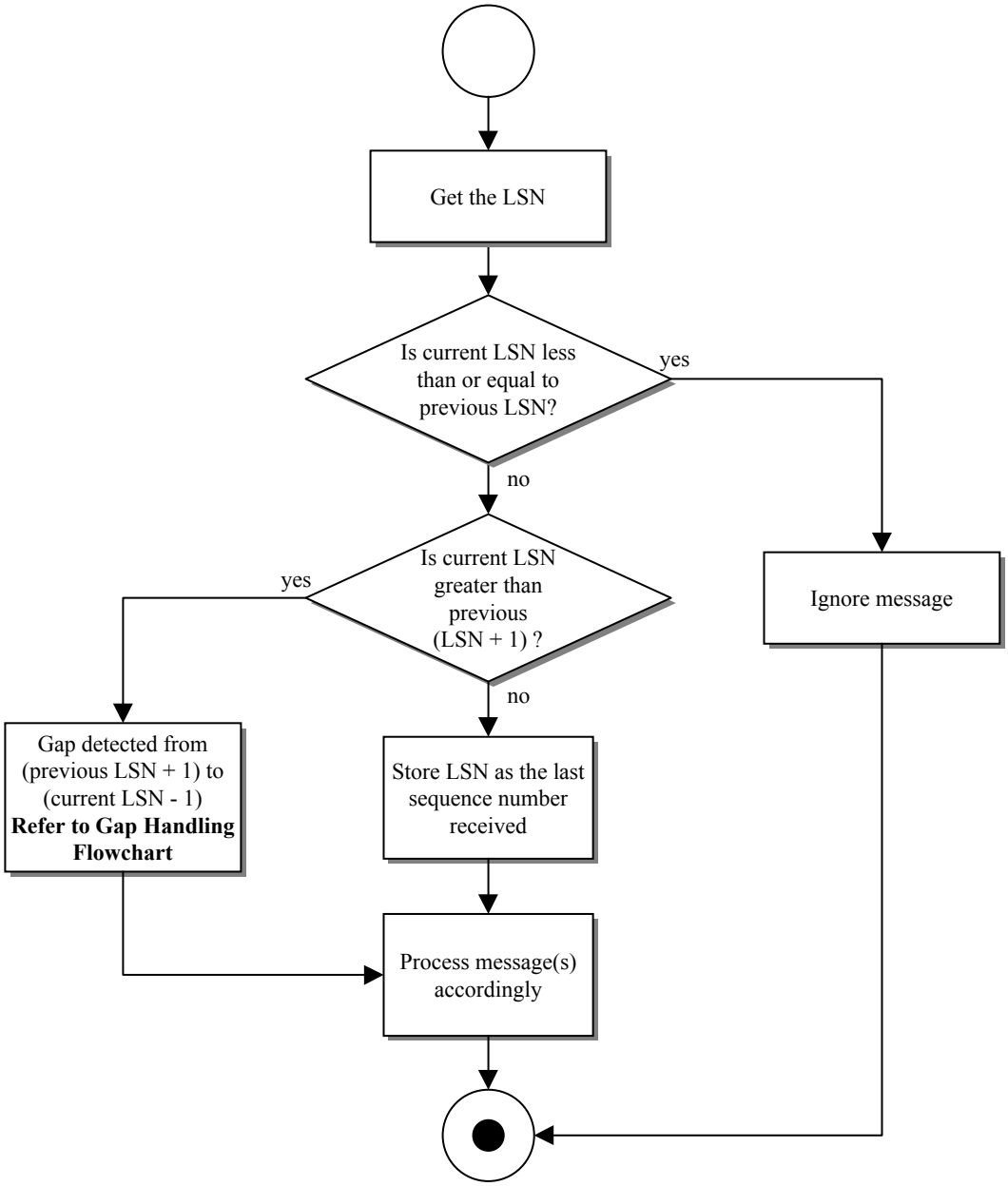

Figure 4. Processing of Data Messages

Processing of Gap Handling

The following is the recommended way of handling gaps:

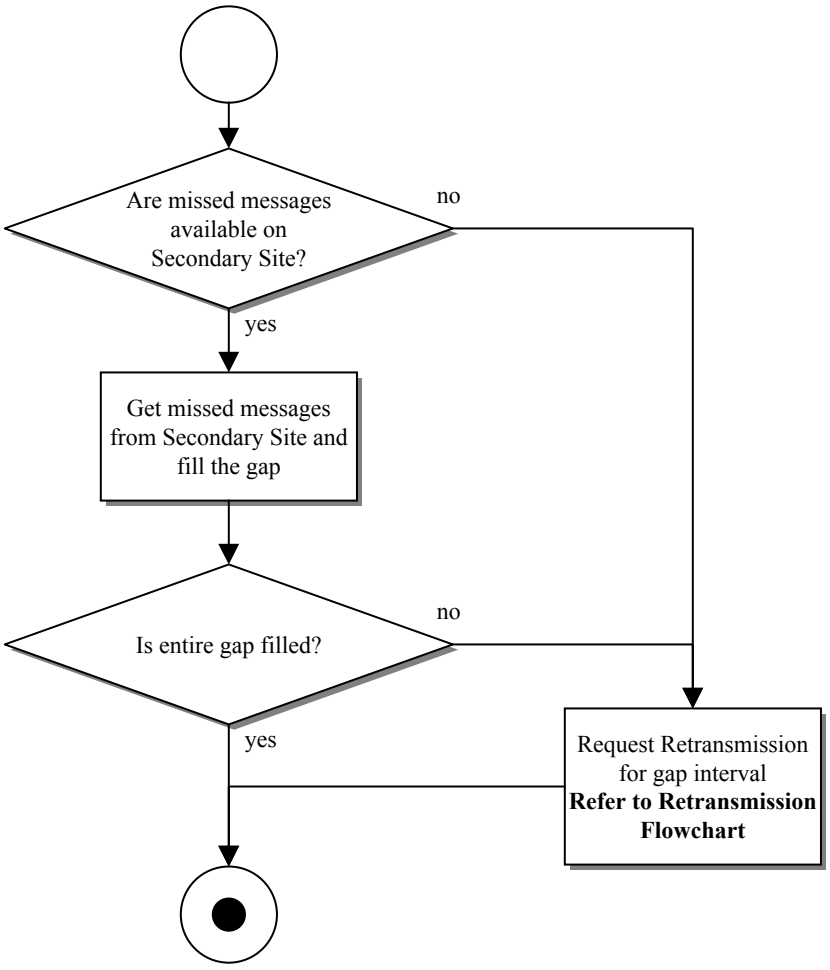

Figure 5. Processing of Gap Handling

Processing of Line Level Retransmissions

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

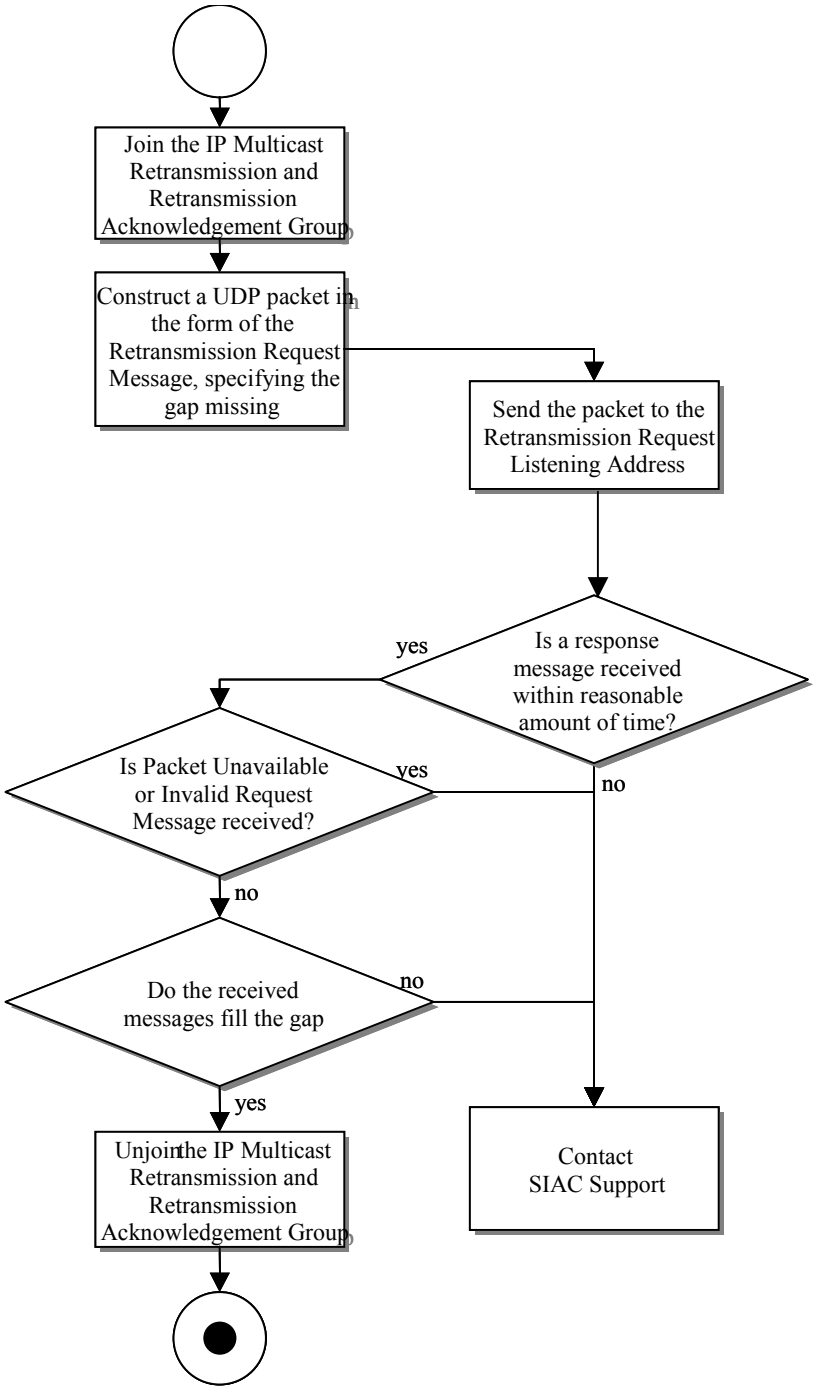

Figure 6. Processing of Line Level Retransmissions

Processing of Update Messages at Symbol Level

The following is the recommended way of processing messages at symbol level:

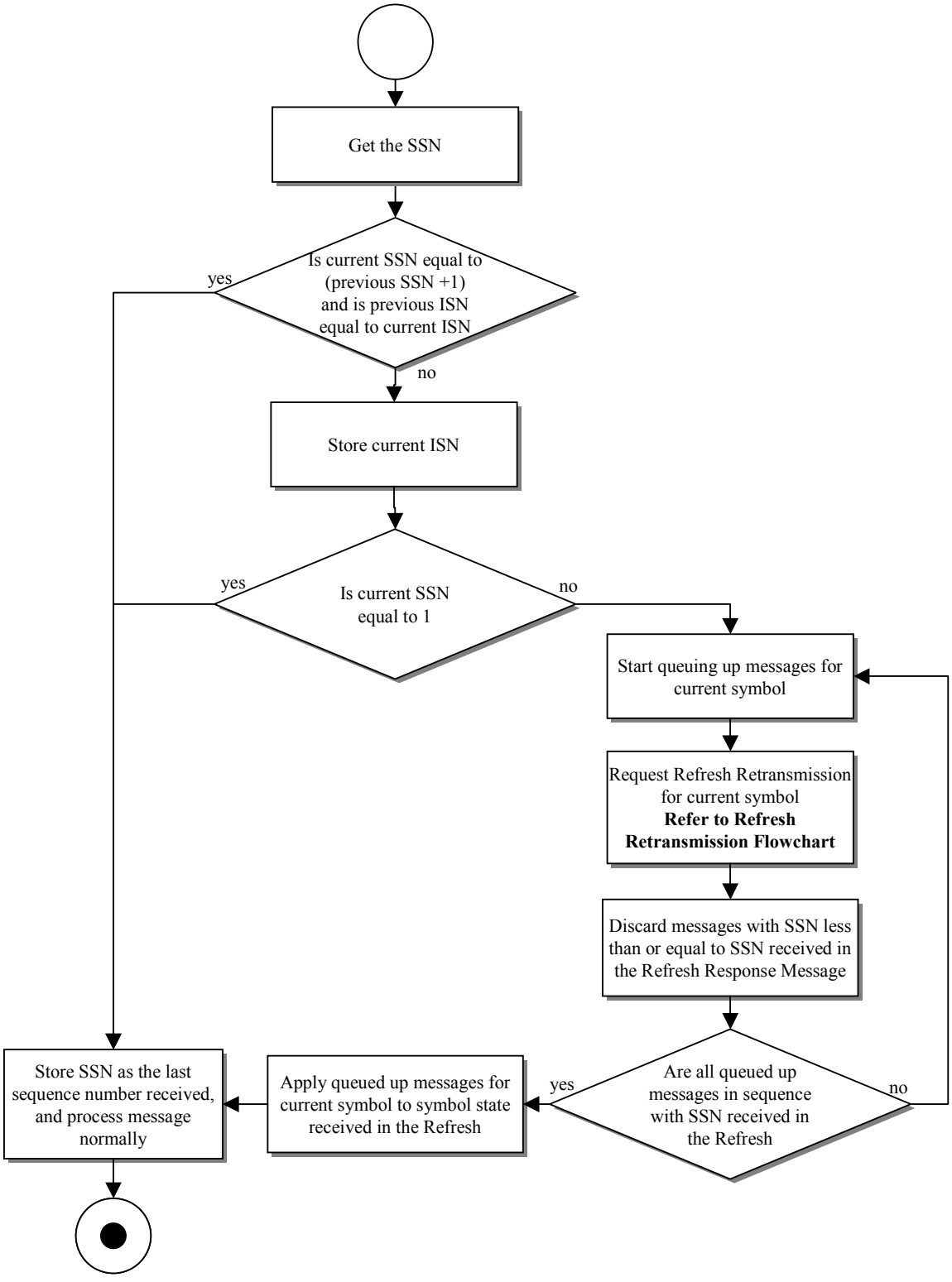

Figure 7. Processing of Update Messages at Symbol Level

Processing of Refresh Retransmissions

The following is the recommended way of processing Refresh Retransmissions at symbol level:

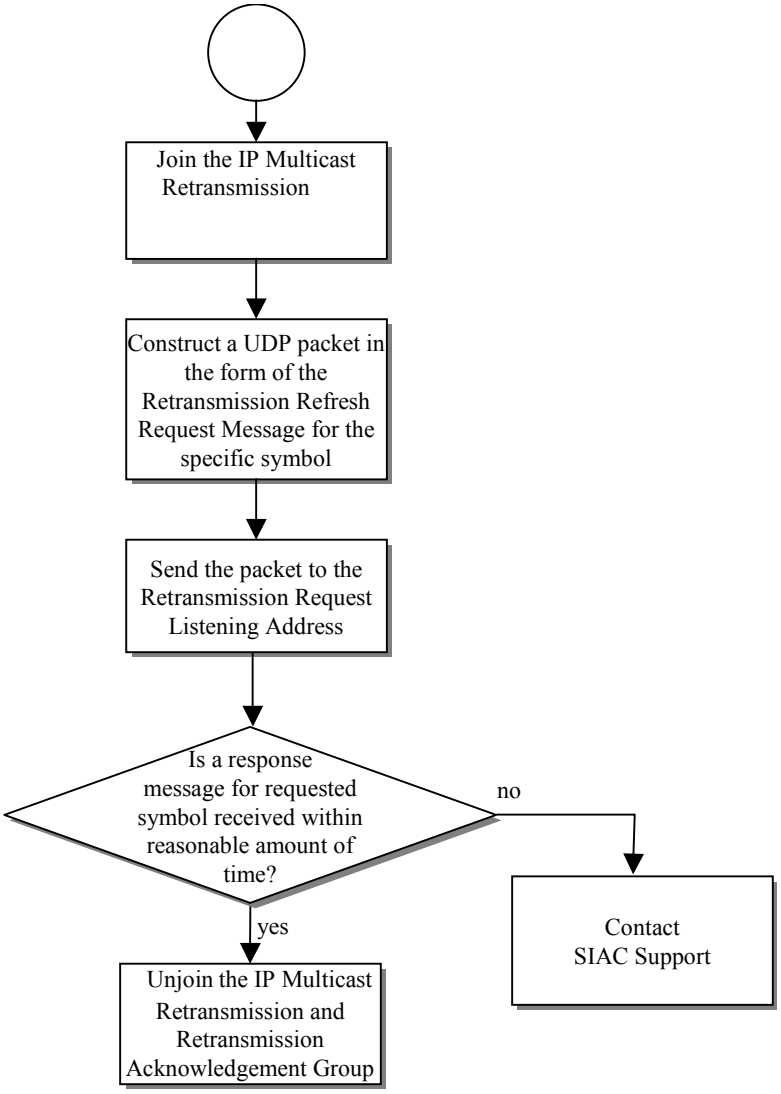

Figure 8. Processing of Refresh Retransmissions

APPENDIX C

This section provides information to assist Subscribers with frequently asked questions concerning the NYSE OpenBook (Real-Time Updates)feed.

Frequently Asked Questions

Q: What is the average message size?
A: Average message size is approximately 95 bytes. However, keep in mind that we will send all the price points as part of a Full Update Message. This message can span multiple packets and be very large.

Q: What is the maximum message packet size?
A: Currently, the maximum message packet size is 1500 bytes.

Q: What is the process to define my Source ID for retransmission purposes?
A: Contact SIAC Support and provide the desired Source ID (under the constraints of the format defined in Table 10. Generic Retransmission Request Message Format). SIAC Support will evaluate and approve or disapprove the Source ID. In case of disapproval, a new Source ID has to be defined. In case of approval, SIAC Support will make the necessary updates on the product provider side to add the Source ID and applicable rules.

Q: What is the average message rate (messages per second) seen in a normal day for NYSE OpenBook (Real-Time Updates)?
A: Refer to the NYSE OpenBook (Real-Time Updates) Impact Guide.

Q: What is the average number of messages seen in a normal day?
A: Refer to the NYSE OpenBook (Real-Time Updates) Impact Guide.

Q: 1500 bytes appear small for a max UDP datagram size of 15k; why is it done this way?
A: In order to avoid fragmentation at the network level, the message size is limited to the size of the network MTU, which is 1500 bytes (which includes PDP message plus the IP overhead).

Q: Can the size of the MTU be increased because it is one to one communication between a Subscriber and the SFTI network?
A: The MTU cannot be increased because there are multiple routers before the point where Subscribers connect to the SFTI network, and other Subscribers who may have their network MTU set to a lower value.

Q: Will retransmitted data ever come down the normal data feed?
A: No, retransmitted data will always be sent out on the designated retransmission IP/Multicast address/port.

Q: Are the Primary and Secondary feeds identical?
A: The feeds are not identical because they are distributed from 2 (two) different source addresses. However, the data content (like the sequence numbers and message content) are the same and can be used to fill gaps.

Q: We continue to see gaps in the feed even though our network is isolated and our server is underutilized. What could it be?

A: Although collisions are very rare, it is possible to have message gaps due to them. However, it is more likely that your multicast receiver is gapping during a message burst. This may be due to a UDP buffer overflow. SIAC recommends that Subscribers increase the standard UDP buffer setting to capture this burst.

Q: If it is possible for Symbol Sequence Numbers (SSN) to reset, then it is difficult to handle a gap since it is unclear if the SSN applies to the message I know vs. one I might have missed. How should this be handled?

A: Whenever a SSN is lower than the last expected, you can assume a reset occurred in between. We recommend using ISN field in the message to determine if you can apply this message to the last known one for the symbol. Every time a reset occurs the ISN changes. Therefore, the message can be applied to last known state for the symbol only if current ISN is equal to ISN of previous message for the same symbol.

Known Issues

The following issues have been found in the feed and are being addressed by future updates to the service.

Q: I sometimes see duplicate messages on the feed. The sequence numbers look the same as well as the contents. Is there something wrong on my side?

A: No. There is a case in which a network level activity returns an error during data transmission that causes our system to resend the data. However, the original message still goes through. We recommend that when Subscribers receive duplicate messages that they ignore them.

APPENDIX D

Differences from NYSE OpenBook (5-Second Update)

The table Table 27, below, lists the relevant differences between NYSE OpenBook (Real-Time Updates) and NYSE OpenBook (5-Second Update) from the Client Interface Specification perspective.

LOCATION	DESCRIPTION
Title Page	Change of Product Name, instead of NYSE OpenBook; it's now NYSE OpenBook Real-Time
2.1, 2.4, 6.1, 6.2	Change of Product Name, instead of NYSE OpenBook; it's now NYSE OpenBook Real-Time
4.1	CAP is no longer a valid connection point, SFTI is.
Table 5. Required PDP Message Header Fields Table 6. Sequence Number Reset Message Format Table 7. No Data Available Message Format Table 8. Heartbeat Message Format Table 9. Admin Message Format Table 10. Generic Retransmission Request Message Format Table 11. Invalid Generic Request Message Format Table 12. Retransmitted Message Format Table 13. Message Unavailable Message Format Table 17. Update Message Format. Table 19. OpenBook Refresh Request Message Table 20. OpenBook Invalid Refresh Request Message Table 21. OpenBook Refresh Message	Changed Product ID to NYSE OpenBook Real-Time, 108.
6.1	The update frequency has been included in the description of the NYSE OpenBook Real-Time Message Specification

LOCATION	DESCRIPTION
7.	IP Group Assignments have been updated with NYSE OpenBook (Real-Time Updates)'s Multicast Groups and Retransmission Request IP's and port numbers.
Table 26. NYSE OpenBook (Real-Time Updates) Publication Frequency	Possible change. Should the update frequency update be explicitly defined in this table? Now, the table specifies "x-seconds" as the amount of time between updates.
Frequently Asked Questions	Updated answer to question, "What is the average message rate (messages per second) seen in a normal day for OpenBook?" to reflect NYSE OpenBook (Real-Time Updates) message rates. Updated answer to question, "What is the average number of messages seen in a normal day?" to reflect NYSE OpenBook (Real-Time Updates) message numbers. Changed answer to question, "What is the current bandwidth usage of the OpenBook Real-Time feed?" to reflect NYSE OpenBook (Real-Time Updates) projected bandwidth utilization. Changed answer to question, "Can the same T1 line be used for the XPress, LiquidityQuote, and OpenBook feeds?" to reflect NYSE OpenBook (Real-Time Updates) projected bandwidth utilization.

Table 27 - Differences from NYSE OpenBook (5-Second Update)