Operating Omega ATS and Lynx ATS

QUOTE TRANSFER PROTOCOL (QTP) SPECIFICATION v 1.05

Revision History

Date	Revision	Description of Change
April 15, 2016	1.00	Created
April 27 , 2016	1.01	Edits made to document.
May 18, 2016	1.02	Edits made to document.
June 10, 2016	1.03	Formatting and edits made to document.
June 13, 2016	1.05	Edits made to document for publishing.

Table of Contents

1. Overview

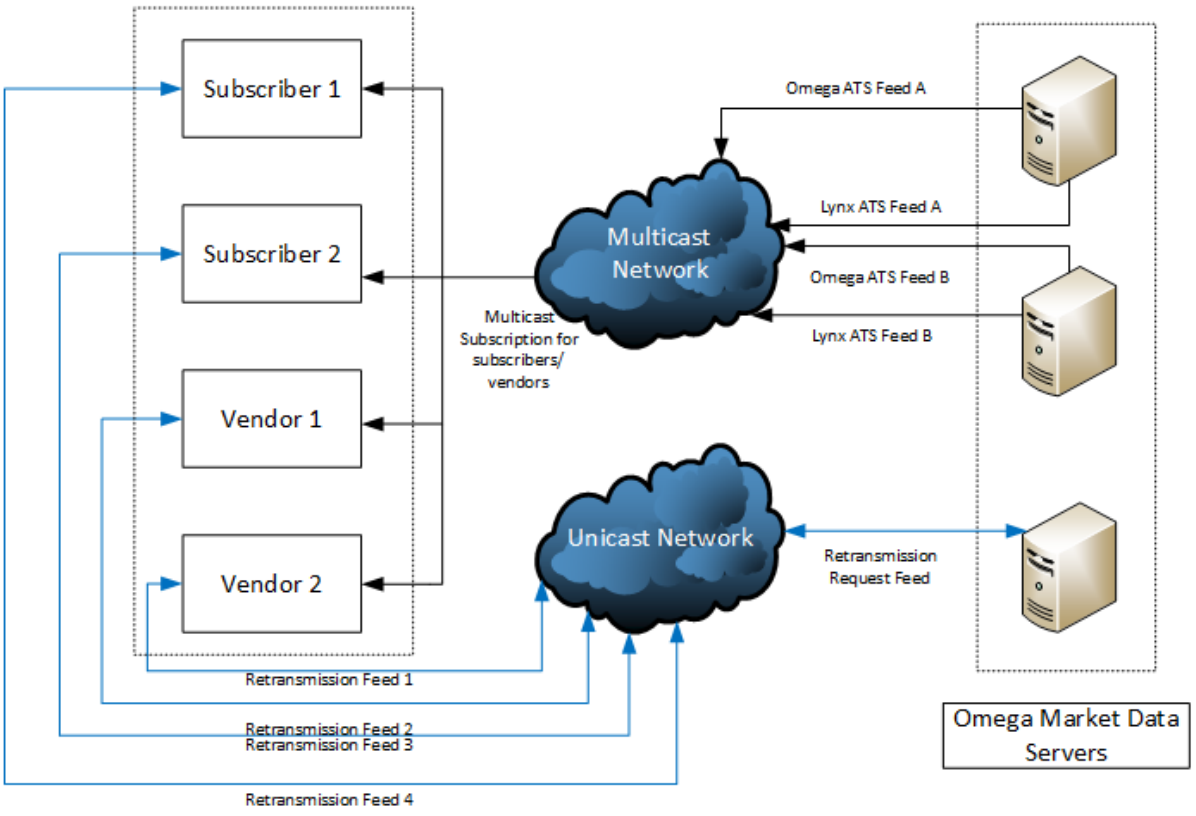
QTP is an efficient and scalable networking protocol for transmitting data messages in a "one transmitter to many listeners" scenario. It is built on top of the UDP protocol and provides a mechanism for users to detect and request missed packets of data.

In QTP, each outbound packet is transmitted only once on each feed, regardless of the number of listeners. One network packet may contain multiple messages to reduce network traffic. This document outlines the messages sent between a QTP server and its clients, while a separate ITCH specification is used to interpret the message payload.

QTP transmitters send downstream packets via UDP multicast to transport the normal data stream to the listeners. These packets are also sent via UDP unicast in response to a Request Message submitted by a listener. QTP clients can send these Request Messages to request the retransmission of any desired packets from the data stream. *OSI allows the retransmission of data for up to 10 minutes prior from the Request Message.*

The QTP server will transmit on a published multicast group for each type of downstream QTP data stream on a network. The listeners must subscribe to this multicast group to receive the downstream data. One or more Request Servers may also be deployed to service any unicast client requests for retransmission of specific messages. The listeners must be configured with these IP addresses and port combinations to which they can submit the requests.

Multicast Architecture



2. Assumptions and Terms

All numeric fields in the QTP messages specified in this document (i.e. sequence number, message counts and message lengths) are binary numbers formatted in Little Endian mode (i.e. least significant byte first). **Note: This does not apply to the data contained the Message Data fields of the Message Blocks.**

Message

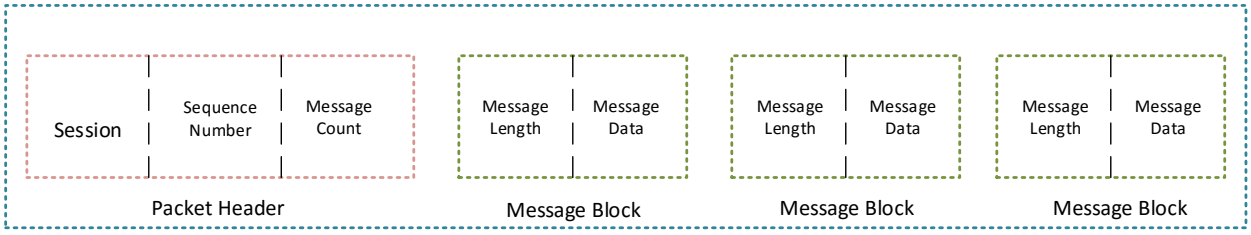
A message is an atomic piece of information carried by the QTP protocol. QTP can theoretically handle individual messages from zero bytes up to 64KB in length although individual messages should be kept small enough so that the underlying UDP network protocol can efficiently carry the resulting QTP packets. The contents of a QTP message are defined by the higher level application (please refer to the ITCH specification).

Session

A session is a sequence of one or more messages. While a single session can last indefinitely, typically the application will define a session to logically group messages together based on time delimitation. Once a session is terminated, no more messages will be sent on that session. A session is considered active if it has started but not yet been terminated.

3. QTP Packet Type

A QTP transmitter sends "downstream" packets that are received by QTP listeners. A QTP packet may contain a payload of 0 or more data stream messages. Each QTP packet consists of a Downstream Packet Header and of a series of Message Blocks. The Message Blocks carry the actual data of the stream.



QTP Packet

3.1 Downstream Packet

3.1.1 Downstream Packet Header

The format of Downstream Packet header:

Field Name	Offset	Length	Value	Notes
Session	0	10	Alphanumeric	Indicates the session to which the packet belongs.
Sequence Number	10	4	Numeric	Sequence number of the first message in the packet.
Message Count	14	2	Numeric	The count of messages contained in this packet.

Sequence Number field:

The Sequence Number field of the packet Header indicates the sequence number of the first message in the packet. If there is more than one message contained in a packet, any messages following the first message are implicitly numbered sequentially.

Message Count field:

The number of Message Blocks contained in a QTP packet is specified by the Message Count field of the Packet Header. The maximum payload size of a Downstream Packet is determined by the sender.

3.1.2 Message Block

The first field of a Message Block is the two bytes Message Length. The remainder of the Message Block is the variable length Message Data field. The first Message Block field will always start immediately following the Header which is 16 bytes from the beginning of the packet. Subsequent Message Blocks will begin after the last byte of the previous Message Block.

The format of Downstream Packet Message Block:

Field Name	Offset	Length	Value	Notes
Message Length	Variable	2	Numeric	Numeric indicates the length in bytes of the message contained in this Message Block.
Message Data	Variable	Variable	Alphanumeric	This is the data payload.

Message Length field:

The Message Length is an unsigned binary count representing the number of message data bytes following this Message Length field.

A Message Length value of zero signals the termination of the session.

Note: A Message Data field does not follow this special Message length value and it must be the last Message block in the packet.

Message Data field:

The Message Data is actual data of the message being transmitted by QTP. For the meaning of the data, please refer to our ITCH specification.

3.1.3 Heartbeat Message

Heartbeat messages are sent periodically by the server so receivers can sense packet loss even during times of low traffic. Typically, these packets are transmitted once per second and contain the next expected Sequence Number. A Heartbeat packet is a QTP packet with a Message Count of zero.

3.1.4 Request Packet

The Request Packet is sent to request the retransmission of a particular message or group of messages. The request packet is sent to a Re-request server. A receiver may need to send this request when it detects a sequence number gap in received messages. The response to a Request Packet is a standard Downstream Packet.

The format of Request Packet:

Field Name	Offset	Length	Value	Notes
Session	0	10	Alphanumeric	Indicates the session to which this belongs.
Sequence Number	10	4	Numeric	First requested sequence number.
Requested Message Count	14	2	Numeric	The number of messages requested for retransmission.

Sequence Number

The Sequence Number field of the packet Header indicates the sequence number of the first message requested.

Requested Message Count

The Message Count indicates how many messages should be retransmitted. If the total size of the requested messages exceeds the maximum payload size of the server, only the number of messages that completely fit will be returned.

4. Receiver Example

A typical QTP receiver client would be configured with the following parameters:

a) The UDP port to listen on and the Multicast group to join.

b) A list of one or more Request Servers those are available to answer retransmission requests for this stream. Each server is specified as a host IP address and a UDP port to which to send requests.

c) A session and sequence number of the next expected message if the client is being restarted.

1. A typical QTP receiver client might obey the following flowchart: Open a UDP socket for the appropriate port and join the desired multicast group.

2. Examine the first received packet to determine the currently active session.

3. If the received session does not match the expected session, abort and report the error.

4. Examine the sequence number of the first recently received packet.

5. If the sequence number does not match the next expected sequence number, send a Request Packet to the Request Server with expected packet number. Wait for a new packet and return to step 4.

6. Process each of the received messages in the packet. If a special "Session Terminated' message is received, handle the End of Session event.

7. Wait for a new packet and return to step 4.

5. Multicast Groups

The Omega data feed is split into two data streams (Feed A & Feed B) for all Omega ATS and Lynx ATS supported symbols.

Below are the multicast information for Omega ATS and Lynx ATS GTE Feeds A & B and Production Feeds A & B.

OMEGA ATS and LYNX ATS TEST MARKET DATA GROUPS & PORTS

Omega ATS TEST Multicast Market Data	Multicast Group	Port
Omega ATS GTE Test Feed A	233.223.59.210	3120
Omega ATS GTE Test Feed B	233.223.59.211	3121

Lynx ATS TEST Multicast Market Data	Multicast Group	Port
Lynx ATS GTE Test Feed A	233.223.59.212	3122
Lynx ATS GTE Test Feed B	233.223.59.213	3123

OMEGA ATS and LYNX ATS PRODUCTION MARKET DATA GROUPS & PORTS

Omega ATS PRODUCTION Multicast Market Data	Multicast Group	Port
Omega ATS Production Feed A	233.223.59.100	3550
Omega ATS Production Feed B	233.223.59.101	3551

Lynx ATS PRODUCTION Multicast Market Data	Multicast Group	Port
Lynx ATS Production Feed A	233.223.59.102	3552
Lynx ATS Production Feed B	233.223.59.103	3553

Rendez-vous Point: 176.67.248.81

Retransmission Server IP and port information to be provided separately by request.

6. Production Bandwidth Requirements:

Marketplace Market Data Feed	Bandwidth Requirements
Omega ATS	24Mb/s
Lynx ATS	12Mb/s