MIAX Options Exchange

TCP Session Management (SesM)

Protocol Specification

Modified 8/13/2015
Version 1.1e

Table of Contents

1. Overview

In an attempt to provide a robust and uniform session management protocol across all interfaces that require guaranteed delivery of certain messages, Miami International Securities Exchange (MIAX) has created TCP Session Management (SesM) protocol. It is a lightweight point-to-point protocol that is built on top of TCP/IP protocol in order to facilitate client/server communication.

SesM features:

Session level authentication: SesM includes a simple protocol that allows the server to authenticate the client on login. This allows the right user to connect without service disruptions that could be caused due to wrong connections allowed by a protocol without authentication.

Quick link failure detection: Uses bi-directional heartbeats to quickly and proactively detect link failures.

Failure recovery: The additional benefit that it provides over raw TCP/IP protocol is that it ensures the delivery of sequenced messages in correct order across TCP/IP sessions even in the event of a reconnection after a connection failure.

Flexibility in sequencing: It provides the flexibility to the application layer to deliver messages as sequenced or unsequenced. Application layer can choose the messages that need to be sequenced to ensure quick recovery. SesM delivers unsequenced messages on a best-effort basis only, meaning that they may be lost in the case of a TCP/IP socket failure.

Clear delineation from Application layer: SesM is designed to be used in conjunction with higher level protocols that specify the contents of the messages that SesM delivers. The SesM protocol layer is opaque to the higher-level messages permitting it to encapsulate and transport an application message consisting of ASCII or binary data.

2. SesM Protocol

2.1 Packets

SesM client and server communicate by exchanging SesM packets. SesM packets do not necessarily map directly to physical packets on the underlying network socket; they may be broken apart or aggregated by the TCP/IP stack. SesM protocol limits the maximum payload length to 65,534 bytes. SesM payload is exactly 1 application message encapsulated in a SesM protocol packet. The payload may contain binary data including the line feed character.

2.2 Messaging

2.2.1 Establishing a Session

A SesM connection begins when a client opens a TCP/IP socket to the server and sends a Login packet. If the login request is valid, the server responds with a login response packet following which, the client and the server can begin sending data packets. The connection continues until the TCP/IP socket is broken or closed.

The sequence number of the first sequenced message in each session is always 1. Typically, when initially logging into a server, the client will set the *Requested Sequence Number* field to 1 and set the *Requested Session* to zero in the login request. The client will then inspect the login response to determine the currently active session. If the TCP/IP connection is ever broken, the client can then re-log into the server indicating the current session and its next expected sequence number. By doing this, the client is guaranteed to always receive every sequenced packet in order, despite TCP/IP connection failures. If the client sends a zero for the next expected sequence number, server will only send new packets generated after login and not send any old packets to the client.

2.2.2 Sending and Receiving Application Messages

Once the client has successfully established a session with the server, the client and the server can start communicating using application messages. SesM encapsulates each application message into a SesM packet. Each data packet carries a single higher-level protocol message. Sequenced data packets contain an explicit sequence number enabling the client and server to be in sync and allowing every packet to be individually identifiable. Unsequenced data packets do not contain a sequence number and are not counted in the sequence numbering used in the sequenced data packets.

2.2.3 Gap Fill

SesM clients can request a gap-fill in one of the following available methods.

- *Login Request*: SesM client sends in the expected sequence number of the next packet in the *Login request*. SesM server sends (from its store) to the client all the Sequenced packets with a sequence number that is greater than or equal to the expected sequence number. SesM server will keep the connection live after this login synchronization is complete. Some application interfaces will not support this method if they have limited number of gap-fill servicing servers that have to be shared by many clients. This is the preferred method for application interfaces that have dedicated client/server connections.
- *Retransmission Request*: SesM clients that have a need for a gap-fill with a specific start and end sequence number must use the *Retransmission request*. SesM server sends all the packets as per the request and disconnects the client connection. This is the preferred method for retransmission of packets that the client application may have missed from a data feed. For SesM clients that must use the Retransmission request, Login message must be sent with a Requested Sequence Number of zero (0).

Note: For retransmission interfaces such as the Top of Market Retransmission Services, SesM clients must use the *Retransmission request* instead of the *Login request* in order to fill gaps.

2.2.4 Heartbeats

SesM uses heartbeat packets to quickly and proactively detect link failures. The server must send a *Server heartbeat* packet anytime more than 1 second has passed since the server last sent any data. This ensures that the client will receive data on a regular basis. If the client does not receive anything (neither data nor heartbeats) for an extended period of time (typically 3 heartbeats), it can assume that the link is down and attempt to reconnect using a new TCP/IP socket.

Similarly, once logged in, the client must send a *Client heartbeat* packet anytime more than 1 second has passed since the client last sent anything. If the server does not receive anything from the client for an extended period of time (typically 3 heartbeats), it can close the existing socket and listen for a new connection.

2.2.5 Ending a Session

The server indicates that the current session has ended by sending an *End of Session* packet. This indicates that there will be no more packets contained in this session.

The client can send a *Logout* packet to request the server to close its connection. The client will have to reconnect and login with the new Session ID or a Session ID of zero to begin receiving messages for the next available session.

2.3 Data Types

The following table describes the data types used in SesM protocol:

Data Type	Description
Binary	Unsigned, little-endian byte-ordered, binary encoded numbers
Alphanumeric	Each byte can contain characters or numbers. Left justified and space-padded on to the right

2.4 Configuration

SesM client/server setup will require the following configuration:

- *Username & Computer ID*: Client will use this in the Login packet and server will only allow connections from a valid clients
- *SesM version*: Client will use this in the Login packet and server will only allow connections from a client if the client is using the SesM version that the server is expecting based on configuration.

3. Packet Types and Packet Structure

3.1 Application Data Packets

3.1.1 Sequenced Data Packet

The Server application can choose to use Sequenced data packets to send all messages that must be recoverable by the client in case of a connection loss. Each Sequenced data packet carries one sequenced message from the higher-level protocol and the Sequenced data packet structure incorporates a sequence number field to identify each packet. SesM protocol also retains all Sequenced data packets at the Server end until the end of each Server session. These factors allow the server application to guarantee messages and the message sequencing across TCP/IP sessions.

Since SesM packets are carried via TCP/IP sockets, the only way logical packets can be lost is in the event of a TCP/IP socket connection failure. In this case, the client can reconnect to the server and request the next expected sequence number and pick up from where it left off.

Sequenced Packet Structure

Packet Length	Packet Type	Sequence Number	Application Data

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet (meaning the total length of the payload, packet type, sequence number)
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "S"** in this field indicates that this is a sequenced data packet
Sequence Number	8	Binary	Sequence number of this sequenced data packet
Application Data	Variable	Any	1 Application Message

3.1.2 Unsequenced Data Packet

Server can choose to send some data in the form of unsequenced data packets; Client can send data to the server only in the form of unsequenced data packets. Each unsequenced data packet carries one unsequenced message from the higher-lever protocol. These messages are unsequenced, and must not update the client's or server's sequence numbers.

Unsequenced data packets are delivered on a best-effort basis only. They will be delivered at most once in the same order that they are generated by the higher level protocol. However, some messages may be lost due to TCP/IP

connection trouble including full buffers. If message loss occurs, no portion of the lost message will be transmitted and SesM does not support the recovery of such lost messages. SesM does not define the higher level protocol behavior when unsequenced messages are lost. These semantics are defined by the higher level protocol. The higher-level protocol must be able to handle these lost messages in the case of a TCP/IP socket connection failure.

Unsequenced Packet Structure

Packet Length	Packet Type	Application Data

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet (meaning the length of the payload plus the length of the packet type)
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "U"** in this field indicates that this is an unsequenced data packet
Application Data	Variable	Any	1 Application Message

3.2 Session Management Protocol Packets

3.2.1 Login Request

The SesM client uses this packet to allow the server to authenticate the client and to convey to the server the last sequenced packet that the client received. The SesM client must send a Login request packet immediately upon establishing a new TCP/IP socket connection to the server. If the login request is not received within a reasonable period of time, the server will terminate the connection. Server will not accept any message on a connection until a successful client login.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "L"** in this field indicates that this is a Login request packet
SesM Version	5	Alphanumeric	1.1 (right padded with spaces)
Username	5	Alphanumeric	Username issued by MIAX during initial setup
Computer ID	8	Alphanumeric	ID issued by MIAX during initial setup
Application Protocol	8	Alphanumeric	Eg: MEI1.0 (right padded with spaces)
Requested Session	1	Binary	Specifies the session the client would like to log into, or zero to log into the currently active session.

Requested Sequence Number	8	Binary	Specifies client requested sequence number • next sequence number the client wants to receive upon connection, or • 0 to start receiving only new messages without any replay of old messages

Notes:

- Client must use the Username and Computer ID agreed upon at the time of setup. These fields provide simple authentication to prevent a client from inadvertently connecting to the wrong server. Both Username and Computer ID are case-insensitive and should be padded on the right with spaces.
- Server will also validate the SesM version number sent in the login request with preconfigured version number and reject the login if the client and server are configured to use different versions of the SesM protocol.
- The server can send a GoodBye packet and terminate a client TCP/IP socket if it does not receive a Login request packet within a reasonable period of time (typically 30 seconds).
- If the value supplied in the field *Requested Sequence Number* is higher than the current sequence number, the Server rejects the login request.
- For retransmission interfaces such as the Top of Market Retransmission Services, SesM clients must use the *Retransmission request* instead of the *Login request* in order to fill gaps.

3.2.2 Login Response

The SesM server sends a Login response packet in response a Login request packet from the client. This packet will always be the first packet sent by the server after a successful login request.

The server closes the socket connection if the login request fails.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "R"** in this field indicates that this is a Login response packet
Login Status	1	Alphanumeric	" " – Login successful "X" – Rejected: Invalid Username/Computer ID combination "S" – Requested session is not available "N" – Invalid start sequence number requested "I" – Incompatible Session protocol version "A" – Incompatible Application protocol version "L" – Request rejected because client already logged in
Session ID	1	Binary	The session ID of the session that is now logged into.
Highest Sequence Number	8	Binary	The highest sequence number that the server currently has for the client.

Notes:

- Clients can use the *Highest sequence number* just as an indication as to when the replay of old messages ends and client is current with the server stream. In case new messages arrive at the Server during the replay of the old messages, Server will have more messages than the *Highest sequence number*. Client will have to wait for the Synchronization complete packet as a confirmation of end of replay.

3.2.3 Synchronization Complete

In case the client sends in a login request with a sequence number that is less than the current sequence number that the server has for the client, the server will send a Synchronization complete packet when it is done replaying all the old packets and is ready to start sending new packets. This indicates to the client that the server and hence the client is current with the stream.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "C"** in this field indicates that this is a Synchronization complete packet

Notes:

- This packet will not be sent if the server does not have to replay any old messages.

3.2.4 Retransmission Request

The SesM client uses this packet to request the server to retransmit a range of packets.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "A"** in this field indicates that this is a Retransmission request packet
Start Sequence Number	8	Binary	Sequence number of the first packet to be retransmitted
End Sequence Number	8	Binary	Sequence number of the last packet to be retransmitted

Notes:

- For retransmission interfaces such as the Top of Market Retransmission Services, SesM clients must use the *Retransmission request* instead of the *Login request* in order to fill gaps.
- At the completion of the retransmission, server will disconnect the SesM client. If the End Sequence number is greater than the last packet that the server has, server will send up to the last packet it has at the time of receipt of the request and then the server will disconnect.
- Only Sequenced application messages can be retransmitted.

- SesM clients must not send heartbeats during gapfill. Otherwise, upon receipt of a heartbeat, the server will reset the connection if it is done sending all the data. This may make the client not read the rest of the data on the connection.

3.2.5 Logout Request

The **client may send** a Logout request packet to request the connection be terminated. Upon receiving a Logout request packet, the server will immediately terminate the connection and close the associated TCP/IP socket.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "X"** in this field indicates that this is a Logout request packet
Reason	1	Alphanumeric	" " – Graceful Logout (Done for now) "B" – Bad SesM Packet "L" – Timed out waiting for Login Packet "A" – Application terminating connection
Text	Variable	Alphanumeric	Free form human readable text to provide more details beyond the reasons mentioned above.

3.2.6 GoodBye Packet

The **server may send** a GoodBye packet to inform the client a reason for termination of the connection just before it terminates the connection. Upon sending a GoodBye packet, the server will immediately terminate the connection and close the associated TCP/IP socket. The client can also immediately close the associated TCP/IP socket.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "G"** in this field indicates that this is a GoodBye packet
Reason	1	Alphanumeric	"B" – Bad SesM Packet "L" – Timed out waiting for Login Packet "A" – Application terminating connection
Text	Variable	Alphanumeric	Free form human readable text to provide more details beyond the reasons mentioned above.

3.2.7 End of Session

The server will send an End of Session packet to denote that the current session is finished. The connection will be closed shortly after this packet, and the user will no longer be able to reconnect to the current session.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "E"** in this field indicates that this is a End of Session packet

3.2.8 Heartbeat

3.2.8.1 Server Heartbeat Packet

The server should send a Server Heartbeat packet anytime more than 1 second passes where no data has been sent to the client. The client can then assume that the link is lost if it does not receive anything for an extended period of time (3 Heartbeat intervals).

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "0"** (zero) in this field indicates that this is a Server Heartbeat packet

3.2.8.2 Client Heartbeat Packet

The client should send a Client Heartbeat packet anytime more than 1 second passes where no data has been sent to the server. The server can then assume that the link is lost if it does not receive anything for an extended period of time (3 Heartbeat intervals).

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "1"** in this field indicates that this is a Client Heartbeat packet

3.2.9 Test Packet

A Test packet can be sent by either side of a SesM connection at anytime. Test packets are intended to provide human readable text that may aid in debugging problems. Test packets should be ignored by both client and server application software.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Packet Length	2	Binary	The length of rest of the packet
Packet Type	1	Alphanumeric	SesM protocol packet type. A **value of "T"** in this field indicates that this is a Test packet
Text	Variable	Alphanumeric	Free form human readable text

Appendix A: Contact List

Please visit MIAX website at http://www.MIAXOptions.com for obtaining most up-to-date contact list and other such information.

Appendix B: Revision History

Revision Date	Version	Author	Description
Oct 07 2011	1.0	Vinay S. Rao	First official release.
Feb 21 2012	1.1	Vinay S. Rao	Retransmission Packet Type renamed to “A”.
Feb 8 2013	1.1a	Vinay S. Rao	Copyright and page footer updated
Apr 19 2013	1.1c	Vinay S. Rao	Copied the information about correct message to use for gap fill from section 3.2.4 (Retransmission Request) to 3.2.1 (Login Request) and 2.2.3 (Gap Fill)
Sep 10 2013	1.1d	Vinay S. Rao	Updated SesM version to 1.1 to reflect production settings.
Aug 13 2015	1.1e	Vinay S. Rao	Notes of Retransmission request: Guidance to not heartbeat during gapfill.

miaxoptions.com