MIAX Pearl Equities Exchange

MACH
Protocol Specification

Revision Date: 01/05/2022
Version 1.2e

7 Roszel Rd | Princeton, NJ 08540

Table of Contents

1. Overview

In order to provide a robust and uniform session level protocol across all interfaces that require distribution of application messages, MIAX Pearl Equities™ has created the MACH protocol.

MACH is a scalable and lightweight multicast based protocol that allows a publisher to publish to many subscribers. It incorporates session level communication and a packet structure that allows subscribers to detect gaps and know the start and end of a message session.

MACH features:

Scalability and Low latency: MACH uses multicast that allows the publisher to publish once and all subscribers to receive it without any degraded performance with addition of new subscribers. This results in a lower latency compared to any fan-out based solutions.

Increased performance: MACH bunches multiple MACH packets into a single UDP packet that reduces I/O calls at both ends allowing for increased I/O efficiency and performance.

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

Link failure detection: Uses frequent heartbeats to enable subscribers to proactively detect link or publisher failures.
Note that MACH protocol does not provide any delivery guarantee. In case the subscriber detects a gap, publisher side applications are generally designed to have a separate retransmission service or redundant feeds. Refer to the publisher side application manual (example: Top of Market Feed) for additional details of availability of such services.

2. MACH Protocol

2.1 Packets

MACH Publisher distributes application data in the form of MACH packets. MACH packets contain a unique sequence number and each MACH channel has a unique multicast group and/or port.

While the MACH protocol limits the maximum payload length to 65,535 – (Header Size) bytes, publishing applications (example: Top of Market Feed) will keep the UDP packet size small and within the length of a standard MTU so as to get the best performance from the underlying infrastructure.

One or more MACH packets may be placed in the underlying Multicast MTU as the following two examples show:

Structure of the payload with one MACH Packet contained in a UDP packet

MACH Header	Application Data

Structure of the payload with N (eg: 3) MACH packets contained in a UDP packet

One or more of such MACH messages can be buffered into a single UDP packet

.

MACH Header	Application Data	**MACH Header**	Application Data	**MACH Header**	Application Data

2.2 Messaging

2.2.1 Start of Session

A MACH session begins when the Publisher sends a *Start of Session* packet to denote that a new session has started. This is always the first packet in a session.

Subscribers must be able to handle mid-day *start of session* messages from Publisher in case the Publisher needs to restart with a lower or higher sequence number.

Every MACH session has a Session Number assigned to it. The *Start of Session* packet and all subsequent packets in the session have the current Session Number. Session Number will be a value of 1 or above. Any packet with a Session Number less than 1 should be ignored.

2.2.2 Heartbeats

MACH uses heartbeat packets allowing subscribers to proactively detect link failures and Publisher failures. The Publisher must send a *heartbeat* packet anytime more than 1 second has passed since the Publisher last sent

any data. This ensures that the subscribers will receive data on a regular basis. If the subscriber does not receive anything (neither data nor heartbeats) for an extended period of time (typically 3 heartbeat intervals), the link or the Publisher may be down.

Heartbeat packets may be sent out before *Start of Session* packet and may have a Session Number = 0. These heartbeat packets may be ignored

2.2.3 Retransmissions

In case the subscriber detects a gap, publisher side applications are generally designed to have a separate retransmission service or redundant feeds. Refer to the publisher side application manual (example: Top of Market Feed) for additional details of availability of such services.

2.2.4 End of Session

The Publisher 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.

Subscribers must be able to handle mid-day *start of session* messages without the dissemination of an *end of session* message for the previous session from Publisher in case the Publisher experiences issues.

2.3 Data Types

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

Data Type	Description
BinaryU	Unsigned, little-endian byte-ordered, binary encoded numbers
BinaryS	Signed, 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 Packet Header Structure

Field Name	Length	Data Type	Notes
Sequence Number	8	BinaryU	Sequence number of this data packet
Packet Length	2	BinaryU	Length of the packet. This includes the header and application data.
Packet Type	1	BinaryU	MACH protocol packet type. Type: 0 = Heartbeat; 1 = Start of Session; 2 = End of Session; 3 = Application Data
Session Number	1	BinaryU	The Session this packet was generated from

Points to note:

- Packet Length in the header may be used by recipients to determine the beginning of the next message. This allows for smoother migration when application protocol is expanded
    - If new messages are introduced that the client does not need, client may hop over the unused message type using the packet length.
    - If new fields are added to the end of a message, clients that do not need these fields do not need to be modified.

3. Packet Type and Structure

3.1 Session Level Protocol Packets

3.1.1 Start of Session

The Publisher will send a Start of Session packet to denote that a new session has started. This is always the first packet in a session.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Sequence Number	8	BinaryU	Set to 0
Packet Length	2	BinaryU	Length of the packet.
Packet Type	1	BinaryU	Value of 1
Session Number	1	BinaryU	Current Session

3.1.2 Heartbeat

The Publisher will send a Heartbeat packet anytime more than 1 second passes where no data has been published. The subscriber can check if the link is lost if it does not receive anything for an extended period of time (e.g.: 3 Heartbeat intervals).

Heartbeat is sent with the sequence number of the last packet sent out. This will also help the subscriber in proactively recognizing if the last packet was lost. Heartbeats are sent alone and not bundled with application messages.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Sequence Number	8	BinaryU	Sequence number of the last application data packet
Packet Length	2	BinaryU	Length of the packet.
Packet Type	1	BinaryU	Value of 0
Session Number	1	BinaryU	Current Session

3.1.3 End of Session

The Publisher will send an End of Session packet to denote that the current session is finished. There will be no more packets disseminated in this session.

Field Description (length is in bytes):

Field Name	Length	Data Type	Notes
Sequence Number	8	BinaryU	Sequence number of the last application data packet
Packet Length	2	BinaryU	Length of the packet.
Packet Type	1	BinaryU	Value of 2
Session Number	1	BinaryU	Current Session

3.2 Application Data Packets

The Publisher application will send all messages as Sequenced data packets. The sequence number is unique for a given MACH session for a given multicast group/port pair. Each Sequenced data packet carries one application message from the higher-level protocol. The subscriber must use the sequence numbers to recognize missing packets (gaps) and out of order packets.

Structure of MACH message containing application data

Sequence Number	Packet Length	Packet Type	Session Number	Application Data

Field Description:

Field Name	Length	Data Type	Notes
Sequence Number	8	BinaryU	Sequence number of this data packet
Packet Length	2	BinaryU	Length of the packet. Header and App Data
Packet Type	1	BinaryU	Value of 3
Session Number	1	BinaryU	Current Session
Application Message	Variable	Any	1 Application Message

Appendix A: Contact List

Please visit MIAX website for obtaining most up-to-date contact list and other such information.

Appendix B: Revision History

Revision Date	Version	Description
Oct 07, 2011	1.0	First official release.
Nov 10, 2011	1.1	Update MACH header
Jun 19, 2012	1.2	Update MACH Header 1) Increase the size of the Sequence number from 4 bytes to 8 bytes 2) Add a one byte Session Number
Feb 08, 2013	1.2a	Updated legal statement in footer
Sep 30, 2016	1.2b	Updated Logo and Copyright info.
Oct 20, 2016	1.2c	Fixed some typos in section 3.2
Mar 22, 2018	1.2d	Clarified that Heartbeats are not bundled with application messages. Noted that the *Start of Session* packet and all subsequent packets will have the current Session Number which is a value of 1 or above. Noted that Heartbeat packets may be sent out before the *Start of Session* packet and may have Session Number = 0
Jan 05, 2022	1.2e	Clarified that Start of Session message has a Sequence Number = 0

miaxglobal.com