In this blog we will discuss about heartbeat and other related attributes of channel.

**Why Heartbeat in MQ?**

Any network failure reported to the MQ channel will cause the channel to end. If there is a network failure and not reported to the channel, the channel will not end, for example, the waiting state of a receiver channel is on a recv() call. Until the recv() call returns, the channel will not end of the connection. Sometimes sender MQI/MCA process does not recognize it in a timely way when the network had actually failed, as the communication protocols i.e.(TCP/IP) doesn’t always return errors immediately.

“**Heartbeat**” is one of the enhancement of MCA/MQI which improves the situation considerably. The heartbeat flow is used to detect whether a MQ channel is still ‘alive’ or not i.e. in other words network Failure.

**Heartbeat:**

The heartbeat flow is used to detect whether a channel is still ‘alive’ or not. For Sender and Receiver channels, heartbeats can flow from both the server side as well as the client side independently. If no data has been transferred across the channel for the **heartbeat interval(HBINT)**, the sender MCA (runmqchl) or MQI client initiates a heartbeat flow and the receiver MCA/MQI process (amqrmppa) responds to it with another heartbeat flow. The Receiver MQI/MCA process(amqrmppaa) is also capable of initiating a heartbeat to the client, again irrespective of the state of the channel. To prevent both server-connection and client-connection MQI agents heart beating to each other at the same time, the server heartbeat is flowed after no data has been transferred across the channel for the heartbeat interval plus 5 seconds.

Heartbeat flow is managed on the basis of the following attribute on the channel.

**Heartbeat interval(HBINT):**

The Heartbeat Interval or **HBINT** is the channel attribute which controls how often the sending end of the channel should check that the receiver is active when there are no messages to send.

The value is in seconds and must be in the range 0 – 999 999. A value of zero means that no heartbeat flows are to be sent. The default value is 300.

There can be three possible ways of setting the **HBINT** attribute on the channel. And following are consequences.

1. **Default value 300 being set on both sender and receiver channel:**

   In the above case, if no message is being sent/received by the sending MCA/MQI process, then after 300 seconds, sending MCA/MQI process will send the heartbeat to the receiver MCA/MQI process.
II. Sender channel HBINT value and the receiver channel HBIT value are not same:
In this case, the negotiated HBINT value will be the higher value of the two.

III. Sender channel HBINT value and the receiver channel HBIT value are set to ‘0’:
A value of zero means that no heartbeat flows are to be sent

Note:
A relatively high value could be chosen as the default so that there would be very little impact on the network. For performance, the Heartbeat value should not be too small. The low heartbeat values will cause no extra network traffic provided the channel remains busy.

Receive Timeout:
It is not the heartbeat interval on its own that decides to close the channel. There is another parameter called Receive Timeout which decides how long to wait for the reply to the heartbeat before making the assumption that the connection is ‘dead’.

After the channels negotiate a HBINT value, if HBINT is set to less than 60 seconds, the receive time-out value is set to twice this value. If HBINT is set to 60 seconds or more, the receive time-out value is set to 60 seconds greater than the value of HBINT.

For example:

If the negotiated HBINT value is 20 seconds then, the receive timeout value will be 40 seconds.

If the negotiated HBINT value is 100 seconds then, the receive timeout value will be 160 seconds.

So, if no data is being received within the receive timeout period i.e. 40 or 160 seconds as per the above example then connection will be closed.

Note:
Canceling the connection after twice the heartbeat interval is valid because a data or heartbeat flow is expected at least at every heartbeat interval. Setting the heartbeat interval too small, however, can cause problems, especially if you are using channel exits.

Example:
If the negotiated HBINT value is one second, and a send or receive exit is used, the receiving end waits for only 2 seconds before canceling the channel. If the MCA is performing a task such as encrypting the message, this value might be too short.

Disconnect interval (DISCINT):
This attribute is the length of time after which a channel closes down if no message arrives during that period. You can specify any number of seconds from zero through 999 999 where a value of zero means no disconnect; wait indefinitely.

The default DISCINT value is set to 100 minutes. A value of a few minutes is often a reasonable value to use without impacting performance or keeping channels running for unnecessarily long periods of time.

When the sender and receiver channel are active but not communicating, to check if the connection is alive or not, heartbeat is being initiated in each HBINT time.
If DISCINT is being set to 10 minutes, and no communication happens for 10 minutes, then the connection will be closed.

**Note:**
The value of DISCINT should always be equal or greater than HBINT.

*Batch heartbeat interval (BATCHHB):*
This attribute allows a sending channel to verify that the receiving channel is still active just before committing a batch of messages.

The value is in milliseconds and must be in the range zero through 999999. A value of zero indicates that batch heart beating is not used.


The batch interval allows us to back-out rather than being in-doubt, if the receiver is not active. By backing out the batch, the messages remain available, so that they could be redirected to another channel. If the sending channel has had a communication from the receiving channel within the batch heartbeat interval, the receiving channel is assumed to be still active, otherwise a ‘heartbeat’ is sent to the receiving channel to check. The sending channel waits for a response from the receiving end of the channel for an interval, based on the number of seconds specified in the channel Heartbeat Interval (HBINT) attribute.
To determine whether this batch heartbeat flow is sent, we note the time we last received a flow from our partner. If this time was longer ago than the interval specified in the Batch Heartbeat or BATCHHB parameter on the channel, then we will send the flow, otherwise we will not. The interval specified is therefore a reflection of the stability of your network.

*Keepalive Interval:*
TCP/IP also has its own heartbeat protocol, called KeepAlive, allowing it to recognize network failures. The use of KeepAlive is recommended.

This attribute is used to specify a timeout value for a channel. The value is ignored for all channels that have a Transport Type (TRPTYPE) other than TCP or SPX.

The Keepalive interval attribute is a value which passes to the communication stack specifying the Keepalive timing for the channel. The value indicates a time, in seconds, and must be in the range 0 – 99999. A Keepalive Interval value of 0 indicates that channel-specific Keepalive is not enabled for the channel and only the system-wide Keepalive value set in TCP/IP is used.

On system level, it occurs when the KEEPALIVE=YES parameter is specified in the TCP stanza in the distributed queueing configuration file, qm.ini, or through the IBM® MQ Explorer. Keepalive must also be switched on within TCP/IP itself, using the TCP profile configuration data set.

You can also set the Keepalive interval to auto: if Keepalive is set to auto then, 60 seconds will be added to the negotiated HBINT.

The following Technote shares the detailed description [here](#)
For example:
If negotiated HBINT is 60 seconds, then Keepalive interval will be 120 seconds.
The use of heartbeats and receive time-out remove the need for KeepAlive when both ends
support both options, however KeepAlive can still be used as well. When communicating
with MQ implementations which do not support both heartbeats and the non-blocking reads,
you should still use KeepAlive. Keepalive is strongly recommended for SVRCONN channels
since even if client heartbeats are available, they are only used during MQGETs.

Contributor: Prateek Kulkarni