

Minimizing data loss when using MQTT Store and Forward

When MQTT Transmission Store and Forward is enabled and the edge node detects a disconnect with the MQTT Server, tags will be stored locally in a history store.

If your system is fully Sparkplug B compliant, you can also use Store and Forward to ensure that data is buffered locally in the event the main application is offline but the client side is still connected to the server. The MQTT Sparkplug B specification defines a way for subscribing clients (such as MQTT Engine) to notify publishing clients (such as MQTT Transmission) via a STATE message when they are ONLINE or OFFLINE. The publishing clients can then key off a particular subscribing clients status to do different things when that subscriber status changes either to offline or online. MQTT Transmission uses this functionality to enable the store and forward of tag changes within the module.

How quickly the edge node detects these conditions and starts to store the data will minimize the data loss when using MQTT Store and Forward and there are a couple of configuration parameters within MQTT Transmission to assist with this: Keep Alive and Primary Host ID.

MQTT Transmission

Keep Alive

The MQTT protocol is hosted on top of the TCP protocol, which is connection-oriented, and provides a stable and orderly flow of bytes between two connected parties. Normally when the MQTT Transmission client attempts to send data over TCP, it will receive an error at the TCP level if disconnected and can immediately start to buffer data locally.

However, in some cases, TCP can have half-connection problems. A half-connection is a connection that has been disconnected or not established on one side, while the connection on the other side is still maintained. In this case, the half-connected party may continuously send data, which obviously never reaches the other side. To avoid black holes in communication caused by half-connections, the MQTT protocol provides a Keep Alive mechanism that allows the client and MQTT server to determine whether there is a half-connection problem, and close the corresponding connection.

For MQTT Transmission the Keep Alive is the maximum interval in seconds between any two MQTT protocol control packets sent by the MQTT Transmission client to the server. If the client is idle and has no control packets to send, it will send PINGREQ protocol packets and expects to receive PINGRESP packets from the server. If no PINGRESP is received within 1.5 times the Keep Alive interval, the client will close the connection.

This means that, depending on the TCP connection failure, it can take up to 1.5 times the Keep Alive interval for MQTT Transmission to determine that it is disconnected and start to buffer data locally at the Edge and during this time data will be lost.

For example, a Keep Alive of 30 seconds could result in up to 45 seconds of data loss in the event of a TCP half-connection scenario. Reducing the Keep Alive to 5 seconds would correspondingly lower the potential data loss to 8 seconds.



Setting the Keep Alive to a short interval will increase the data traffic over the TCP connection in the event that the client is idle.

Setting the Keep Alive to 0 will disable the Keep Alive functionality

Primary Host ID

When [Primary Host ID](#) is configured, MQTT Transmission will subscribe on STATE notification topics. If MQTT Transmission is notified that the primary backend application, configured in the Primary Host ID, is 'OFFLINE', it will close its client connection with the MQTT server and walk to the next MQTT server defined in the set. Data will be stored locally until the state of the primary host is "ONLINE" when any stored data will be published.



The notion of Primary Host ID is part of the Sparkplug B specification and, as such, both the main application and edge nodes in a system will need to be Sparkplug compliant

MQTT Engine

If you are using MQTT Engine as your subscribing client you can configure the ability for publishing clients to be notified when its is ONLINE or OFFLINE. The configuration parameters to do this are: Keep Alive, Primary Host Enabled and Primary Host ID.

Keep Alive

For MQTT Engine the [Keep Alive](#) is the maximum interval in seconds between any two MQTT protocol control packets sent by the MQTT Engine client to the server. If the client is idle and has no control packets to send, it will send PINGREQ protocol packets and expects to receive PINGRESP packets from the server. If no PINGRESP is received within 1.5 times the Keep Alive interval, the client will close the connection.



Setting the Keep Alive to a short interval will increase the data traffic over the TCP connection in the event that the client is idle.

Setting the Keep Alive to 0 will disable the Keep Alive functionality

Primary Host ID

For MQTT Engine when [Primary Host Enabled](#) is true and Primary Host ID is configured, MQTT Engine will publish it's connection state on a topic that contains the Primary Host ID.

In the event that MQTT Engine becomes disconnected from the server, the server will publish a LWT message on the same topic, setting the state to 'OFFLINE'. Any connecting client that subscribes on the state topic will then be notified of the MQTT Engine state.

This means that, depending on the TCP connection failure, it can take up to 1.5 times the Keep Alive interval for the server to determine that MQTT Engine is disconnected and publish this LWT. During this time data from Edge Node devices will be lost.

For example, a Keep Alive of 30 seconds could result in up to 45 seconds of data loss in the event of a TCP half-connection scenario. Reducing the Keep Alive to 5 seconds would corresponding lower the the potential data loss to 8 seconds.