Determining the settings for an MQTT Transmission History Store for pre 4.0.19 modules

Introduction

MQTT Store and Forward allows data to be buffered locally at a client when connections are down to the MQTT Server infrastructure and deliver that data when the connection is restored. This feature is critical in most applications because if we lose connection to the MQTT Server we will lose data if it is not buffered locally. When Store and Forward is enabled and the edge node detects a disconnect to the MQTT Server, tags will be stored locally in a history store. When the edge node can reconnect to the MQTT Server, it will publish any stored tags. The rate at which these stored tags are flushed from the history store is configurable to prevent any delays in the delivery of live data.

Determining the settings for the MQTT Transmission History Store requires understanding the unique system properties at each Edge Node. There are a number of factors involved in determining how large the tag capacity can be including but not limited to system resources such as CPU, RAM (especially when using 'In-Memory'), Disk IOPS (if using 'Disk-Backed'), the nominal tag change rate (e.g. number of tags changing per second in the system), the flush rate, bandwidth availability, whether flushing in order vs asynchronously, etc.

> ✓ Review the MQTT Transmission Transmitters and Tag Trees documentation for help in identifying your Edge Node(s) and Device(s).
>
> If using UDTs you should include each member tag of a UDT in your count when determining the number of tags.

MQTT Transmission History Store requires the following parameters:

- **Type**
  - The type of History Store with options of In-Memory and Disk-Backed
  - Data stored in an In-Memory History Store will not be persisted across a module configuration change, module disable/enable, module restart or power loss.
  - Data stored in a Disk-Backed History Store will persist across a module configuration change, module disable/enable, module restart or power loss.
- **Edge Node Tag Capacity**
  - Maximum number of Edge Node level tag change events to store per Edge Node before dropping oldest historical Edge Node tag change events. This value is independent of the 'Device Tag Capacity' and only applies to how many 'Edge Node level' tag change events are stored per Edge Node.
  - Default 100,000
  - A tag change event is triggered by either a change in value or quality and results in the tag's Qualified Value (which has three attributes of value, quality and timestamp) being stored.
- **Device Tag Capacity**
  - Maximum number of Device tag change events to store per Device before dropping oldest historical Device tag change events. This value is independent of the 'Edge Node Tag Capacity' and only applies to how many 'Device level' tag change events are stored per Device.
  - Default 10,000
  - A tag change event is triggered by either a change in value or quality and results in the tag's Qualified Value (which has three attributes of value, quality and timestamp) being stored.
- **Flush Quantity**
  - The maximum number of tags to publish in a single message upon reestablishing communication.
  - Default 10,000
- **Flush Period**
  - The period to wait in milliseconds between publishes when flushing messages upon reestablishing communication.
  - Default 200

> ⓘ Large Tag Capacities
>
> If using a large tag capacity it is highly recommended to test the system under load in a non-production environment on similar hardware and software that will be used in production. During testing is also important to get the system into a state where the store and forward cache becomes full before beginning to flush. This will ensure that the system is sized appropriately when deployed into a production environment. Generally any capacity over 2,000,000 is considered large and should be tested before deploying to production.

How to determine these settings

Testing is the best way to determine your settings for your Edge Gateway(s) and the recommended approach is to:

- Configure a History Store with large Edge Node Tag Capacity and Device Tag Capacity.
- Disconnect for a set time period
- Monitor the live updates for the count of tag change events and memory used from the Transmission Info History Store metrics.
- Reconnect and monitor the data flushing to ensure that the flush settings (quantity and period) are able to handle the current tag changes that continue to build.

> ⚠ Whilst the history flush is in progress, all new change events are written to the history store until it has been completely flushed. If the tag change rate at the Edge is faster than the MQTT Transmission Flush Period this can cause a build up of data in history store(s) and prevent the publishing of live data.

- Extrapolate based on the test

> ✓ For ease of control we recommend setting a Primary Host ID in both MQTT Engine and in the MQTT Transmission.
>
> Removing the Primary Host ID from MQTT Engine will simulate the primary backend application going offline and cause MQTT Transmission to store data to the History Store.
>
> Restoring the Primary Host ID to MQTT Engine will simulate that the primary backend application is online and cause MQTT Transmission to flush the stored data.

> ⚠ The History Store values used in the video were selected to easily show the process for testing. It would be expected that a much higher Flush Quantity would be required in a live environment to stop a build up of data in history store preventing the publishing of live data.