

Connections to AWS IoT oscillating

AWS IoT has a message size limit of 128KB and will disconnect the client if it receives a message that exceeds this limit.

As a result, messages published from MQTT Transmission or MQTT Engine that exceed the limit can cause connection to the broker to oscillate.

The two likely causes of large messages are:

- MQTT Sparkplug BIRTH and DEATH message lengths
- Slow tag pacing

MQTT Sparkplug BIRTH and DEATH message lengths

As required by the [Sparkplug Specification](#) data is published on the are the NBIRTH, NDEATH, DBIRTH and DDEATH topics.

The payloads for each of these messages include every metric the device will ever report on. At a minimum these metrics must include the name, datatype and current value.

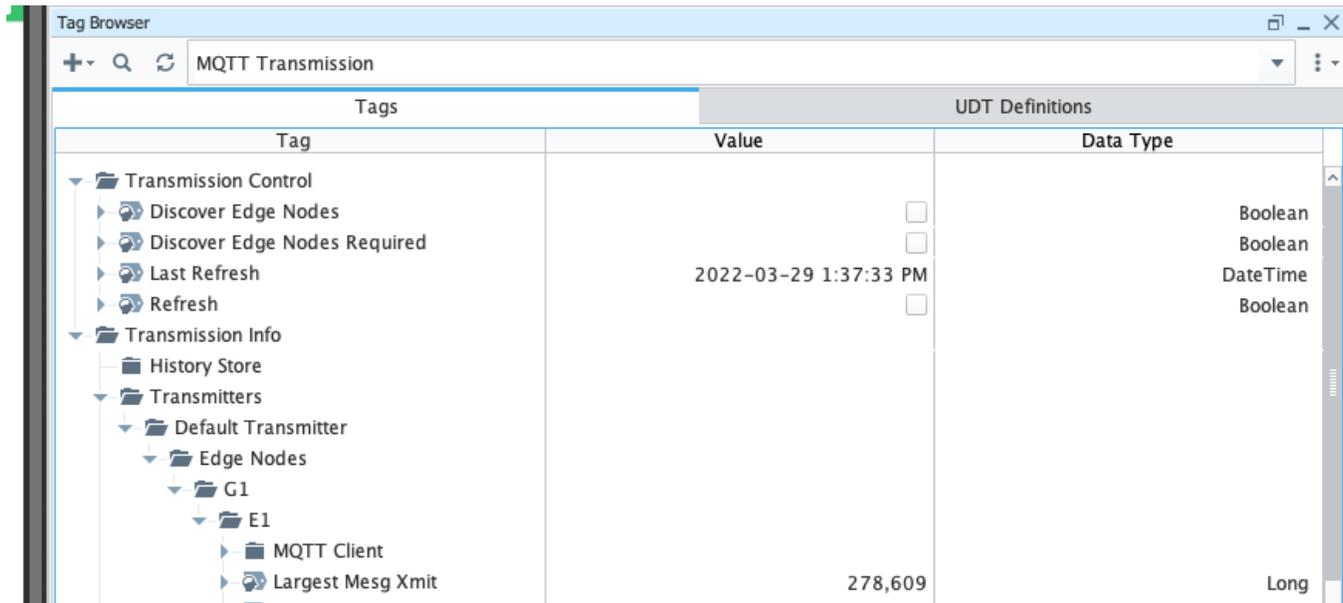
If you have large number of tags in a single folder, these messages can easily exceed the 128KB limit.

Slow tag pacing

Tag pacing is the buffer period for outgoing Transmission messages which allows tag change events to be buffered to reduce the number of generated MQTT messages. If you set this pacing too slow, with a large volume of tag changes, messages can easily exceed the 128KB limit.

How to find your largest message size

To find the largest published message size, navigate to the MQTT Transmission in the Ignition Designer Tag Browser. Drill down to your Transmitter and you should see the 'Largest Mesg Xmit' tag as shown below.



The screenshot shows the Ignition Designer Tag Browser interface. The window title is 'Tag Browser'. The search bar contains 'MQTT Transmission'. The interface is divided into two main sections: 'Tags' and 'UDT Definitions'. The 'Tags' section is expanded to show a tree view of the MQTT Transmission structure. The 'UDT Definitions' section is currently empty. The 'Tags' section shows a tree view with the following structure:

- Transmission Control
 - Discover Edge Nodes (Boolean)
 - Discover Edge Nodes Required (Boolean)
 - Last Refresh (DateTime: 2022-03-29 1:37:33 PM)
 - Refresh (Boolean)
- Transmission Info
 - History Store
 - Transmitters
 - Default Transmitter
 - Edge Nodes
 - G1
 - E1
 - MQTT Client
 - Largest Mesg Xmit (Long: 278,609)

Resolution

- Configure the Transmitter to use a [compression](#) algorithm to compress the payload before publishing
- Modify your tag structure such that no messages exceed this limit.
 - This typically means dividing tags up across 'Device level folders' so the size of each NBIRTH, NDEATH, DBIRTH or DDEATH message is less than 128KB
- Check that your [Tag Pacing](#) period is not causing messages greater the 128KB to be generated.
 - Wireshark can be helpful in performing this tuning.

Unable to Resolve?

If the troubleshooting tips did not help you resolve your issues, please open a ticket with [Support](#) making sure to include the MQTT Engine or MQTT Transmission logs as appropriate.

From the Ignition Logs view, select the Download icon to download a copy of the system-name.idb file to your local file system. You will need to compress (zip, 7z or rar) this file before sending to Support.

Additional Resources

- Inductive Automation's Ignition download with free trial
 - [Current Ignition Release](#)
- Cirrus Link Solutions Modules for Ignition
 - [Ignition Strategic Partner Modules](#)
- Support questions
 - Check out the Cirrus Link Forum: <https://forum.cirrus-link.com/>
 - Contact support: support@cirrus-link.com
- Sales questions
 - Email: sales@cirrus-link.com
 - Phone: +1 (844) 924-7787
- About Cirrus Link
 - <https://www.cirrus-link.com/about-us/>