B: Sparkplug Payload Compression

Sparkplug B Payload Compression

Some Sparkplug libraries contain utilities and helpers for doing compression and decompression of Sparkplug B payloads. Alternatively this compress and decompression can be done by any code or application.

Here are the steps required to apply compression to a Sparkplug B payload before publishing it to an MQTT Server.

1. Select a compression algorithm.

   While there are many compression algorithms to choose from, we currently support:
   a. DEFLATE (ZLIB RFC 1950)
   b. GZIP

2. Create the payload and encode it using Google Protocol Buffers.

   This results in a typical encoded array of bytes that would normally (no compression) be published to an MQTT Server.

3. Compress the payload bytes using one of the above algorithms.

4. Create a new payload to wrap the compressed bytes.

   This new payload should have UUID set to "SPBV1.0_COMPRESSED" and the same timestamp and sequence number as the original compressed payload. The compressed bytes should be stored in the "body" field of this new payload. A single metric should be added to the new payload to specify the algorithm used to compress the original payload. If not "algorithm" metric is present, it will be assumed that the DEFLATE algorithm was used.

   EXAMPLES

   a. Example: Payload compression with no algorithm specified (defaults to DEFLATE)

```
{
    "uuid" : "SPBV1.0_COMPRESSED",
    "timestamp" : 1486144502122,
    "seq" : 0,
    "body" : <compressed-encoded-payload>
}
```

   b. Example: Payload compression with DEFLATE algorithm

```
{
    "uuid" : "SPBV1.0_COMPRESSED",
    "timestamp" : 1486144502122,
    "seq" : 0,
    "body" : <compressed-encoded-payload>
    "metrics" : [ {
        "name" : "algorithm",
        "dataType" : "String",
        "value" : "DEFLATE"
    } ]
}
```

   c. Example: Payload compression with GZIP algorithm

```
{
    "uuid" : "SPBV1.0_COMPRESSED",
    "timestamp" : 1486144502122,
    "seq" : 0,
    "body" : <compressed-encoded-payload>
    "metrics" : [ {
        "name" : "algorithm",
        "dataType" : "String",
        "value" : "GZIP"
    } ]
}
```

5. Finally, encode the new payload using Google Protocol Buffers and it is ready to be published.