Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

C. Jennings
S. Nandakumar
Cisco

20 October 2025

Metrics over MOQT
draft-jennings-moq-metrics-02

Abstract

This document specifies how to send metrics type information over the Media Over QUIC Transport (MOQT).

Many systems produce significant volumes of metrics which either are not all needed at the same time for consumption by collection/ aggregation endpoints or may also compete for bandwidth with the primary application, thus exacerbating congestion conditions especially in low-bandwidth networks. Delivering these over architectures enabled by publish/subscribe transport like Media Over QUIC Transport (MOQT) [I-D.ietf-moq-transport], allows metrics data to be prioritized within the congestion context of the primary application as well as enabling local nodes to cache the metric value to be later retrieved via new subscriptions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 April 2026.

Copyright Notice

Table of Contents

1. Introduction

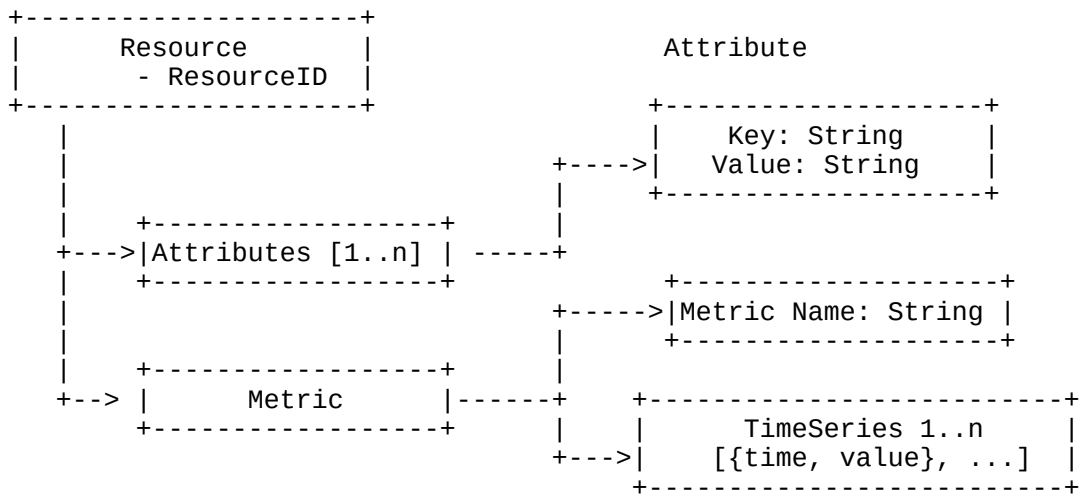
Systems often run into the problem when the network bandwidth for metrics is shared with the realtime media which impacts the media quality. This is especially true for realtime systems wherein metrics compete with bandwidth for media resulting in reduction of the available peak bandwidth for the primary application and often cause congestion in low-bandwidth networks. There is a desire to transport the metrics data at an appropriate priority level over the same transport as the application media. This allows the metrics data to take advantage of times when the media bitrate is below the peak rate while not impacting the peak rate available for media.

Publishing metrics over architectures enabled via MOQT provides additional benefits of leveraging MOQT Relays as caches to store metrics in local nodes. This allows limiting the volume of data to sent to the metrics collectors or aggregator. Instead, metrics that are frequently collected and reported can be cached in local nodes and a new subscription can fetch only the metrics that are needed. This allows for a "just in time" delivery of metrics.

This document specifies how to send metrics type information over the Media Over QUIC Transport (MOQT).

2. Metrics Data Model

Below picture captures relationship between the entities in the data model.



TODO: Define ABNF.

The devices or systems publishing the metrics are referred to as "Resources" and have an unique "ResourceID".

Each metric reported by an Resource has an associated timeseries. A metric's timeseries is time ordered set of values (observations at points in time) for the corresponding metric. Metric names are represented as strings and may contain ASCII letters, digits, underscores, and colon. A metric's value can be one of "Gauge" or "Counter". The "Gauge" value type represents a scalar value that always represents the current value being measured. The "Counter" value type is a cumulative value that represents a single monotonically increasing value that can increase or be reset to zero on restart. The Gauge and Counter can represent either a 64-bit floating point value or a 64-bit integer value.

Resources also define zero or more Attributes. Attributes capture the dimensional data to identify any given combination of them for the metrics reported by the resource. Attributes are represented as key-value pairs and represented as strings.

The data model specified in this section is consistent with the "OpenTelemetry Specification 1.34.0" [otel] for metrics when at rest (also known as Timeseries model). See section Section 3 for streaming model where MOQT is used for transporting the metric data.

2.1. ResourceID

Each resource that creates metrics has a unique "ResourceID". This is created by taking the MAC address of the primary network interface in binary, computing the sha1 hash of it, and truncating to lower 64 bits. Note the sha1 does not provide any security properties, it is just a hash that is widely implemented in hardware. If this is not possible, any other random stable 64 bit identifier may be used. The advantage of us MAC address is that many other management systems use this address and it makes it easier to correlate. The disadvantage is that it reveals the MAC address.

3. Metrics and MOQT

This section maps the metrics data model defined in this specification to MOQT Object model (as defined in section 2 of [I-D.ietf-moq-transport]). The core idea is, the URLs for the MOQT objects are setup such that a subscriber could subscribe to each resource reporting metrics separately, and could pick the right metrics level in the subscriptions.

The Track Namespace consisting of following tuples (moq://metrics.moq.arpa/v1/),(resourceID) is defined in this specification. The track name identifies the granularity level for the metrics being published by the resourcce. Thus, a track name can be identified with the tuple (<granularity-level>) and the full track name having the following format (when represented as a string):

moq://metrics.moq.arpa/metrics-v1/<resourceID>/<level>

Following granularity levels are defined in this specification, along with their associated track names.

1. Emergency : 0
2. Alert : 1
3. Critical : 2
4. Error : 3
5. Warning : 4
6. Notice : 5
7. Informational : 6
8. Debug : 7

Mapping of metrics and its reporting frequency at the aforementioned levels by a resource is application defined. However, it can be typical of an implementation to report metrics that represent aggregation of values over larger time intervals or that represent erroneous conditions, to use granularity that is one of Emergency, Alert, Critical and Error. Such metrics have characteristics of being less frequent and hence consume lesser bandwidth. Metrics that are captured more frequently and capture detailed view of system being measured typically are reported at granularity level of Warning and above. Such metrics, however, consumes high bandwidth when published.

The MOQT group ID identifies point in time when a given set of metrics were captured by the resource. Group ID, thus represents capture time as number of milliseconds since "1 Jan 1972" using NTP Era zero conventions and truncated to 62 bit integer. The first object, with MOQT object ID of 0 captures 2 pieces of information:

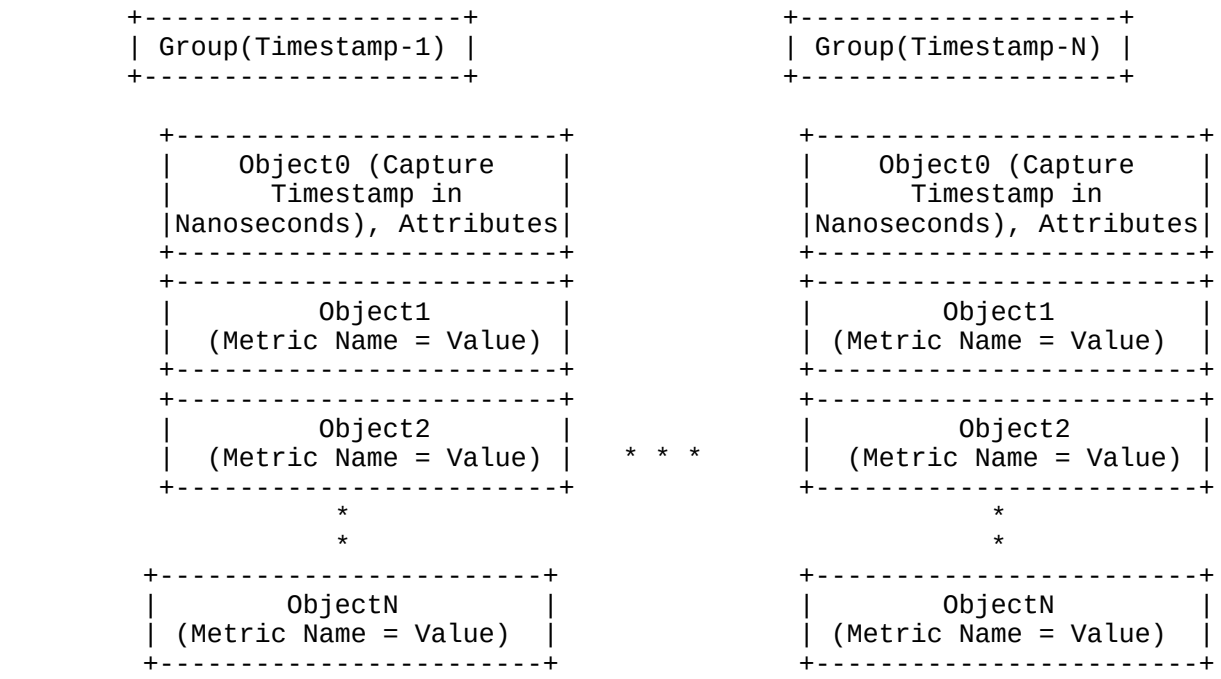
1. The capture timestamp as UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January 1970.

2. One or more attributes scoped to a given resource specified in the track name. This field is optional and if omitted, the attribute values correspond to the most recent object 0 that had any attribute values.

The subsequent objects (Object ID 1 and so on) each capture a metric name and the corresponding value observed at the time provided in the object with the object ID of 0. The metric name field is optional and if omitted, the metric name from the object with same object ID seen in the most recent group is considered.

Below is a conceptual representation of the MOQT mapping for a resource whose resourceID is "resource-1" and granularity level of "warning (4)".

TrackName: resource-1/4



3.1. Examples

Here is an example of Object ID 0 data when represented in JSON format [RFC8259].

```
Group 1, Object ID 0
{
  "capture_timestamp": 1720367991,
  "attributes": [ { "location": "us-east-2"}, {"os": "ubuntu20.4" }]
}
```

Here is an example of Object ID 1 and 2 data showing cpu_usage_percent and cpu_temperature metric values represented in JSON format [RFC8259].

```
Group 1, Object ID 1
{
  "metric_name": "cpu_usage_percentage",
  value: 99
}

Group 1,  Object ID 2
{
  "metric_name": "cpu_temperature",
  "value": 45.1
}
```

Below is another example that shows Group 2 data as continuation from the previous examples, where the redundant information is omitted.

```
Group 2, Object ID 0
{
  "capture_timestamp": 1720369102, // attributes are not repeated
}

Group 2, Object ID 1
{
  "value": 78 // metric_name corresponds to cpu_usage_percentage
}

Group 2, Object ID 2
{
  "value": 21, // metric_name correspons to cpu_temperature
}
```

See Section 2 for details on data types representations for capture timestamp, attributes, metric names and values.

4. Normative References

[I-D.ietf-moq-transport]
Nandakumar, S., Vasiliev, V., Swett, I., and A. Frindell, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-14, 2 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-14>.

5. Informative References

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[otel] "", <https://opentelemetry.io/docs/specs/otel/metrics/data-model/>.

Appendix A. Acknowledgments

Thanks to TODO for contributions and suggestions to this specification.

Authors' Addresses

Cullen Jennings
Cisco
Canada
Email: fluffy@iii.ca

Suhas Nandakumar
Cisco
United States of America
Email: snandaku@cisco.com