

Buffering Messages During Reconnect Attempts

The Core NATS client libraries try as much as possible to be fire and forget, and you should use JetStream functionalities to get higher qualities of service that can deal with Core NATS messages being dropped due to the server connection being interrupted. That said, one of the features that may be included in the library you are using is the ability to buffer outgoing messages when the connection is down.

During a short reconnect, the client can allow applications to publish messages that, because the server is offline, will be cached in the client. The library will then send those messages once reconnected. When the maximum reconnect buffer is reached, messages will no longer be publishable by the client and an error will be returned.

Be aware, while the message appears to be sent to the application it is possible that it is never sent because the connection is never remade. Your applications should use patterns like acknowledgements or use the JetStream publish call to ensure delivery.

For clients that support this feature, you are able to configure the size of this buffer with bytes, messages or both.

Go

```
// Set reconnect buffer size in bytes (5 MB)
nc, err := nats.Connect("demo.nats.io", nats.ReconnectBufSize(5*1024*1024))
if err != nil {
    log.Fatal(err)
}
defer nc.Close()

// Do something with the connection
```

Java

```
Options options = new Options.Builder()
    .server("nats://demo.nats.io:4222")
    .reconnectBufferSize(5 * 1024 * 1024) // Set buffer in bytes
    .build();
```

```
Connection nc = Nats.connect(options);

// Do something with the connection

nc.close();
```

JavaScript

```
// Reconnect buffer size is not configurable on NATS JavaScript client
```

Python

```
# Asyncio NATS client currently does not implement a reconnect buffer
```

C#

```
// Reconnect buffer size is not configurable on NATS .NET client
```

Ruby

```
# There is currently no reconnect pending buffer as part of the Ruby NATS client
```

C

```
natsConnection      *conn      = NULL;
natsOptions          *opts      = NULL;
natsStatus           s          = NATS_OK;

s = natsOptions_Create(&opts);
if (s == NATS_OK)
    // Set reconnect buffer size in bytes (5 MB)
    s = natsOptions_SetReconnectBufSize(opts, 5*1024*1024);
if (s == NATS_OK)
    s = natsConnection_Connect(&conn, opts);

(...)

// Destroy objects that were created
natsConnection_Destroy(conn);
natsOptions_Destroy(opts);
```

As mentioned throughout this document, each client library may behave slightly differently. Please check the documentation for the library you are using.

☐☐ #1

☐ cocobook-sync ☐☐ 2026-10-28 11:12:31 UTC

☐ cocobook-sync ☐☐ 2026-10-28 11:12:31 UTC