



Morse Micro  
reaching farther™

# TCP Keepalive Offload

## APPNOTE-21

May 2024

Author: Nev Young

Copyright © 2024 - Confidential

# Table of Contents

|                              |          |
|------------------------------|----------|
| <b>1 Scope</b>               | <b>3</b> |
| <b>2 Theory of Operation</b> | <b>4</b> |
| 2.1 Other Considerations     | 4        |
| <b>3 How to Configure</b>    | <b>5</b> |
| <b>4 How to Verify</b>       | <b>7</b> |
| <b>5 Revision History</b>    | <b>8</b> |

# 1 Scope

The TCP keepalive offload feature provides the ability for the firmware to support TCP keepalive functionality for an active TCP session. This includes both responding to keepalives from the remote TCP endpoint and also to generate TCP keepalives as required. The main aim is to keep the TCP session active without requiring action from the local host's TCP stack.

The feature is not specific to Standby mode and is available on both HaLow station and AP modes, but the intended use is for station mode specifically.

The TCP keepalive offload feature is available in MM6108 Software Release Package 1.12.x onward.

## 2 Theory of Operation

The feature assumes control of all TCP keepalive operations on behalf of the local TCP stack to make sure the remote endpoint does not drop the connection, particularly when the host is suspended. It doesn't process other TCP packets, apart from snooping them to gather the latest values for things like sequence numbers, so it can process and generate keepalive frames correctly. It needs to have at least one valid TCP exchange processed by the local host before it can start to operate. That could be a keepalive or data exchange.

The basic operation is the feature will respond to TCP keepalives from the remote endpoint. It will only generate a TCP keepalive if nothing is seen from the remote endpoint during the configured period.

The host is required to enable and disable the feature when the TCP session is established (or as needed) and when the session completes (or the feature is no longer needed) via a `morse_cli` command. When enabling the feature the addresses for the TCP endpoints (both local and remote) need to be provided.

If all TCP keepalive retries fail and nothing is received from the remote TCP endpoint then the feature will wake the host with an event identifying that the TCP connection has been lost. An API needs to be agreed with the customer to pass the TCP connection lost event back to the host application managing the TCP connection.

One TCP session at a time is supported.

### 2.1 Other Considerations

For proper operation it requires the local host's TCP stack to be configured with large keepalive timeouts, so the host will not start to generate keepalive frames of its own.

When using the feature with Standby mode, the related offload features that ensure the association and relevant ARP entries are maintained need to be used.

## 3 How to Configure

The command to configure the feature is:

Unset

```
morse_cli tcp_keepalive [-h] [-p <period>] [-c <retry count>] [-i  
<retry interval>] [-s <src IP>] [-d <dest IP>] [-S <src port>] [-D  
<dest port>] {enable|disable}
```

Configure TCP keepalive offload parameters

-h, --help display this help and exit

-p <period> Period in seconds - range 1-65535

-c <retry count> Number of retries - range 0-255

-i <retry interval> Seconds between retries - range 1-255

-s <src IP> Source IP address in dotted decimal notation

-d <dest IP> Destination IP address in dotted decimal  
notation

-S <src port> TCP source port - range 1-65535

-D <dest port> TCP destination port - range 1-65535

{enable|disable} enable/disable TCP keepalive offload

The parameters are defined below:

- **Period** - the time in seconds between locally generated TCP keepalive packets
- **Retry Count** - the number of retries to be attempted before deciding the connection is lost
- **Retry Interval** - the number of seconds between retries
- **Src IP** - the IP address of the remote TCP endpoint (will be the source IP address for received TCP packets)
- **Src Port** - the port number of the remote TCP endpoint
- **Dest IP** - the IP address for the local TCP endpoint
- **Dest Port** - the port number for the local TCP endpoint
- **enable** - activate the feature
- **disable** - deactivate the feature and clear all TCP session information

To enable the feature you need to specify the source and destination IP addresses and ports. Only the TCP stream matching those values will be processed for TCP keepalives. In this context, source refers to the addresses of the remote TCP endpoint and destination is for the local endpoint.

Below are two examples of setting up TCP keepalive:

- `morse_cli tcp_keepalive -p 5 -c 4 -i 2 -s 192.168.1.1 -S 5001 -d 192.168.1.2 -D 45003 enable`
  - Enables the feature with a keepalive period of 5 seconds, retries every 2 seconds and the maximum number of retries is 4. The TCP stream will be for local IP address 192.168.1.2 and local port 45003 and remote IP address 192.168.1.1 and remote port 5001.
- `morse_cli tcp_keepalive -s 10.57.23.7 -S 5050 -d 10.57.23.45 -D 37003 enable`
  - Enables the feature with the default period, etc settings. The TCP stream will be for local IP address 10.57.23.45 and local port 37003 and remote IP address 10.57.23.7 and remote port 5050.

To disable the feature simply use the command:

```
morse_cli tcp_keepalive disable
```

## 4 How to Verify

Several statistics are provided to help identify if the feature is working:

Unset

```
# morse_cli stats -a | grep "TCP keepalive"
```

```
TCP keepalives answered: 6147
```

```
TCP keepalives to host: 33
```

```
TCP keepalives generated: 468
```

```
TCP keepalive connection lost: 1
```

### TCP keepalives answered

The number of TCP keepalive ACK frames sent from the feature. This shows the feature is receiving TCP keepalives from the remote TCP endpoint and is responding.

### TCP keepalives to host

This counter increments when keepalives are forwarded to the host during synchronisation with the TCP session, but may increment faster than the number of keepalives actually forwarded. Once keepalive offload is operational, this value should not increase.

### TCP keepalives generated

The number of TCP keepalive exchanges initiated by the local TCP keepalive feature. If this increments it means TCP keepalives have not been received from the remote side within the configured period.

### TCP keepalive connection lost

This increments if the feature fails to get a response after exhausting the keepalive retries. If the chip is in Standby mode it will wake the host.

## 5 Revision History

| Release Number | Release Date | Release Notes    |
|----------------|--------------|------------------|
| 01             | 02/05/2024   | Initial release. |

**Approvers:** Chad O'Neill (VP of Applications), Matthew Forgie (Director of Software Applications)



**Morse Micro Pty. Ltd. Corporate Headquarters**

Level 8, 10-14 Waterloo Street, Surry Hills, NSW 2010,  
AUSTRALIA

**Morse Micro Inc. – USA**

40 Waterworks Way  
Irvine, CA 92618, UNITED STATES

**Morse Micro – China**

Floor 7, Room 08-210, Wework office 292 Yan'an Rd,  
Shangcheng District, Hangzhou 310003

**USA:** +1.949.501.7080

**Australia:** +61.02.905.45922

**China:** +86.571.229.30277

**Email:** sales@morsemicro.com

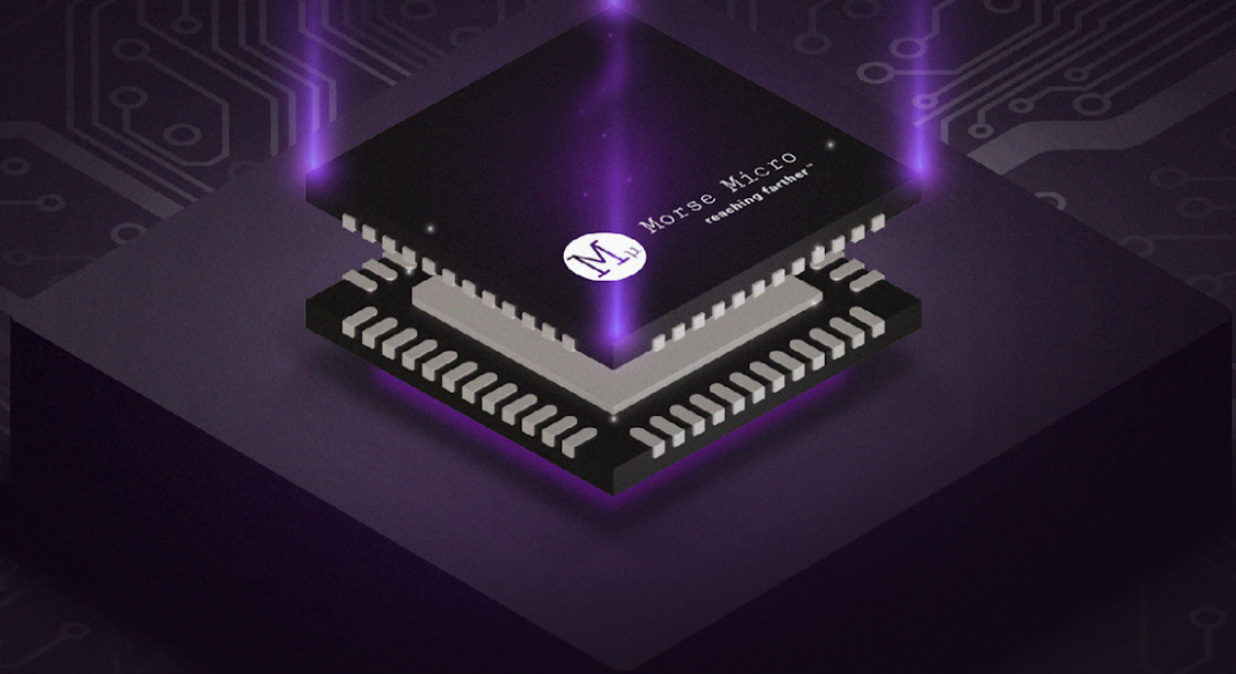
[www.morsemicro.com](http://www.morsemicro.com)

© 2023 Morse Micro Pty. Ltd.

All rights reserved. Morse Micro and the Morse Micro logo are trademarks of Morse Micro Ltd. All other trademarks and service marks are the property of their respective owners.

**About Morse Micro**

Morse Micro is producing IEEE 802.11ah / Wi-Fi HaLow solutions for Internet of Things (IOT) - based on a newly certified Wi-Fi standard called HaLow. Morse Micro is a VC-backed Startup headquartered in Sydney, Australia. Learn more at [www.morsemicro.com](http://www.morsemicro.com)



Morse Micro  
**reaching farther™**