



Morse Micro
reaching farther™

Morse Micro Rate Control

APPNOTE-28

Copyright © 2025 - Confidential

Table of Contents

1 Scope	3
2 History	3
3 Overview	5
4 Details	6
4.1 Initialization	6
4.2 Statistics	6
4.2.1 General	6
4.2.2 Collision Data	7
4.3 Sampling	8
4.4 Decision Logic	8
4.4.1 General	8
4.4.2 Volatile Conditions Detection and Handling	9
4.5 Applying Rates	9
5 Statistics	11
6 Revision History	15

1 Scope

This document covers the design of the Morse Micro Rate Control (MMRC) system. It covers the key mechanisms used by MMRC to choose the best rate and provide a set of rates for the driver to apply to transmit packets.

MMRC selects the rates specifically for unicast data packets.

2 History

Originally Morse Micro used the Minstrel rate control algorithm supplied with the mac80211 driver in the linux kernel. This algorithm works reasonably well, however it was decided to write our own algorithm for the following reasons:

1. In testing Minstrel was not working optimally for 802.11ah. The expectation is that the rate algorithm should be able to get very close to the best fixed-rate throughput at each attenuation point. See the comparison in Figure 1.
2. Minstrel doesn't support the 802.11ah rate MCS 10.
3. Minstrel is slow to react to changes in the environment.
4. Designing a new rate control algorithm gives Morse Micro the ability to customize and improve the rate control algorithm beyond what Minstrel provides.

3 Overview

MMRC controls the MCS, bandwidth and guard interval used for the transmission of packets. It uses recent statistics to determine the PER for various rates and from that decides which rate is best for the highest throughput. The statistics are averaged over time to minimize the impact of minor fluctuations in conditions on the rate.

The rates are applied to each packet at the driver, before they are passed to the chip for transmission. The firmware honors the rates set and does not modify them (unless fixed rates are set in the firmware).

Alternate rates, close to the current best rate, are sampled periodically to check if another rate could be better in the current conditions.

4 Details

4.1 Initialization

When a device associates a new set of rate structures are assigned in MMRC and initialized. The range of rates, bandwidth, and guard intervals allowed (from the S1G capabilities of both devices) are used to determine all the available rates for this association.

Also at this time the initial best rate is selected. It is chosen based on the maximum bandwidth of the association and the averaged RSSI of the received association frames. The initial best rates are set as follows

Average RSSI (dBm)	Initial Best Rate			
Association BW	1MHz	2MHz	4MHz	8MHz
-70 or more	MCS 7	MCS 7	MCS 7	MCS 7
-71 to -85	MCS 3	MCS 3	MCS 3	MCS 3
-86 or less	MCS 3	MCS 3	MCS 0	MCS 0

Table 1 - Initial Best Rates

4.2 Statistics

4.2.1 General

Statistics are gathered from each transmit packet. The number of attempts and successes for each rate are tallied over a rate cycle period (100ms). These are then processed each rate period to generate the exponential weighted moving average (EWMA) of the percentage of successful attempts. This is done for each rate currently in use. The EWMA is used to debounce the statistics to avoid changing rates due to temporary volatility.

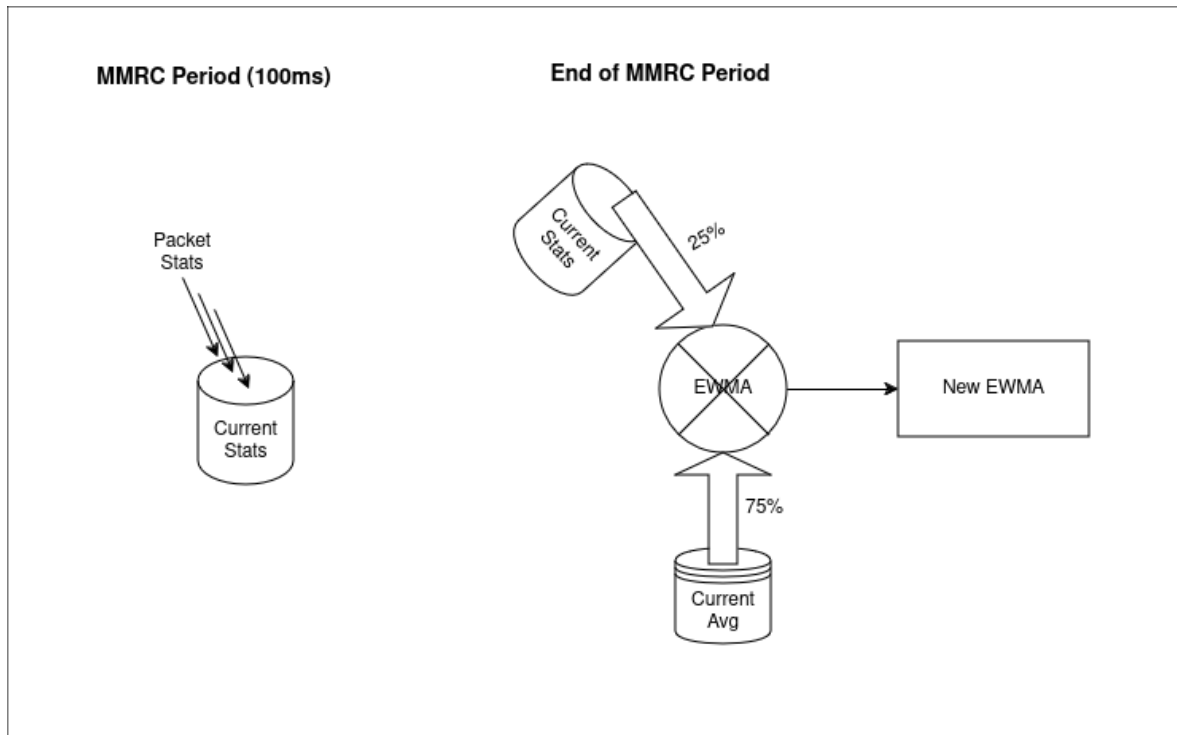


Figure 2 - Statistics Processing

The statistics are usually collected and processed over the 100ms rate period, however if traffic volumes are not sufficient, the statistics will continue to accumulate over more rate periods until a minimum number of statistics is collected for that rate.

4.2.2 Collision Data

As much as possible MMRC avoids collecting statistics for packet attempts that most likely failed due to collisions rather than the rate itself. This includes collisions due to non Wi-Fi/802.11ah as well as competing 802.11ah transmissions. It does this by using only statistics from acknowledged A-MPDUs as long as there are at least some successful MPDUs at that rate. So if the whole A-MPDU is lost MMRC ignores the failed MPDUs for the purpose of its statistics as long as some A-MPDUs do receive a block ACK. That way we don't penalise a rate because of collisions.

Dropping rates while there are collisions is only likely to result in more collisions as the lower rate packets have larger airtime. Additionally, as the receiver can decode the rate, so the issue doesn't lie with the rate itself.

Obviously, if all packets at that rate fail, then we use all the packet statistics so we avoid using that rate.

4.3 Sampling

In order to improve the best rate to higher a MCS in favourable scenarios, MMRC samples other rates. Basically, other rates close to the current best rate are attempted periodically. While doing this the best rate is used as the first retry rate to minimise disruption if the sampled rate is no good.

Sampling is done periodically, with a high 1 in 5 sampling frequency used immediately after association (to help rates converge faster) and a much slower (1 in 50 or 100) used after convergence is achieved.

Generally higher rates are sampled, as lower rates are naturally attempted when retries are required, which mostly covers the case where a lower best rate is optimal. The rate to sample within these parameters is chosen randomly.

Sampling will try other guard interval settings and if the MCS is low it will try lower bandwidths as well.

4.4 Decision Logic

4.4.1 General

The choice of best rate is made by looping through all the rates with recent statistics, calculating a simplified "expected throughput" per rate and comparing them to pick the rate with the highest expected throughput.

The “expected throughput” is calculated by taking the success percentage for the rate and multiplying it by the physical data rate. This obviously will overstate the throughput, but works for comparing rates because each rate is equally affected. It also ignores

- packet sizes
- A-MPDU aggregate sizes
- fragmentation
- overheads (preamble, headers, SIFS, backoff, ACKs, etc)

4.4.2 Volatile Conditions Detection and Handling

MMRC detects unstable environments (such as those with sporadic interference) by comparing the current PER for the best rate to the averaged PER. If the current PER varies up and down from the average PER by more than 5% (over time) then we subtract the EWMA of the absolute PER variation from the PER when calculating the “expected throughput”.

This is only done for the best rate since it is the rate normally being used and therefore the rate whose statistics are most affected by the variations.

4.5 Applying Rates

Each packet is allocated a set of rates to use should multiple attempts be required to transmit the packet successfully. In general, a packet is allocated up to 4 different rates. For most packets the rates allocated will be the best rate, then the next MCS down, then the next down from that and finally the MCS down from the 3rd rate. So if the best rate is MCS 7, the rates allocated would be MCSs 7, 6, 5, 4. The bandwidth and guard interval would remain the same in this case.

Entry	MCS	Attempts
1	7	2
2	6	2
3	5	2
4	4	2

Table 2 - Example of a Standard Rate Table

Under normal circumstances, each of these 4 rates would be attempted twice, giving a total number of attempts of 8 for each packet.

As mentioned above in the sampling section, when sampling a new rate the sample rate will be the first attempted, then followed by the best rate. After that the same 3rd and 4th rates as the normal case. So if MCS 5 is the best rate, normally a packet would use MCSs 5, 4 3 and 2. However, if MMRC is sampling MCS 7, the rates attempted would be MCSs 7, 5, 3 and 2. So the rate table for the sampled packet would be:

Entry	MCS	Attempts
1	7	1
2	5	2
3	3	2
4	2	2

Table 3 - Example of a Sampling Rate Table

5 Statistics

MMRC accumulates statistics into debugfs files. Two files are created, they are

1. **mmrc_table**: this contains a readable table of the statistics for each rate.
2. **mmrc_table_csv**: this is the mmrc_table in CSV format.

To generate these files you will need to mount debugfs on the device.

The files are located on the device in the directory:
`/sys/kernel/debug/ieee80211/phy*/morse`

Here is a sample of part of the mmrc_table from MCS 5 to 7

Morse Micro S1G RC Algorithm Statistics:

Peer 0c:bf:74:cb:45:51

bw	guard	evld	rate_sel	mcs#/ss index	airtime TP(max)	TP(avg)	prob l	ast_rty	last_suc	last_att	tot_suc	tot_att	mpdu_suc	mpdu_fail
1MHz	LGI	0		MCS5 /1 40	5800 0.00	0.00	0	0	0	0	0	0	0	0
1MHz	SGI	0		MCS5 /1 41	5216 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	LGI	0		MCS5 /1 42	2640 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	SGI	0		MCS5 /1 43	2374 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	LGI	0		MCS5 /1 44	1240 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	SGI	0		MCS5 /1 45	1115 0.00	0.00	0	0	0	0	0	0	0	0
8MHz	LGI	0		MCS5 /1 46	560 23.40	23.39	100	0	0	0	3	3	0	0
8MHz	SGI	0	P	MCS5 /1 47	503 26.00	25.99	100	0	0	0	5	5	0	0
1MHz	LGI	0		MCS6 /1 48	4360 0.00	0.00	0	0	0	0	0	0	0	0
1MHz	SGI	0		MCS6 /1 49	3921 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	LGI	0		MCS6 /1 50	2000 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	SGI	0		MCS6 /1 51	1798 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	LGI	0		MCS6 /1 52	920 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	SGI	0		MCS6 /1 53	827 0.00	0.00	0	0	0	0	0	0	0	0
8MHz	LGI	0		MCS6 /1 54	440 26.32	26.32	100	0	0	0	3	3	0	0
8MHz	SGI	0	B	MCS6 /1 55	395 31.00	29.24	100	0	0	0	1769	1786	0	0
1MHz	LGI	0		MCS7 /1 56	3840 0.00	0.00	0	0	0	0	0	0	0	0
1MHz	SGI	0		MCS7 /1 57	3453 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	LGI	0		MCS7 /1 58	1760 0.00	0.00	0	0	0	0	0	0	0	0
2MHz	SGI	0		MCS7 /1 59	1582 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	LGI	0		MCS7 /1 60	840 0.00	0.00	0	0	0	0	0	0	0	0
4MHz	SGI	0		MCS7 /1 61	755 0.00	0.00	0	0	0	0	0	0	0	0
8MHz	LGI	0		MCS7 /1 62	360 29.25	29.24	100	0	0	0	4	4	0	0
8MHz	SGI	0	A L	MCS7 /1 63	323 37.37	31.84	98	0	0	0	107288	110173	0	0
1MHz	LGI	0		MCS10/1 64	64000 0.00	0.00	0	0	0	0	0	0	0	0
1MHz	SGI	0		MCS10/1 65	57562 0.00	0.00	0	0	0	0	0	0	0	0

Amount of packets sent: 111111 including: 865 look-around packets

Figure 3 - Example mmrc_table output

Note, in the mmrc_table the look-around rate is the sampled rate.

The table below describes each field

Parameter	Description
bandwidth	The bandwidth of the rate (eg. 1, 2, 4, 8 MHz)
guard	The guard interval of the rate (SGI, LGI)
evidence	An internal metric of 'evidence' which reflects how relevant a rate is depending on how many times it has been attempted recently. Takes values between 0 and 100 with 100 being most relevant.
rate_selection	Indicates what rates have been selected the last time the rate selection was updated. Takes on the following values: A - Rate has the best throughput B - Rate is 1 MCS lower than the best rate C - Rate is 2 MCSs lower than the best rate P - Rate is 3 MCSs lower than the best rate L - Look around (or sampled) rate Note that multiple letters could appear in a single row.
mcs	The MCS index of the rate.
ss	The number of spatial streams of the rate
index	The unique index of the rate into the MMRC table (each unique combination of rate parameters maps to a unique index)
airtime	The transmit time in microseconds at this rate of a packet with the 'default packet size' (9600 bits or 1200 bytes)

Parameter	Description
max_throughput	Maximum throughput seen at this rate in Mbps. (Theoretical max tput * probability percentage)
average_throughput	The average throughput seen at this rate in Mbps (based on theoretical throughput * probability percentage, averaged over time)
probability	Percentage probability that the rate will succeed (integer, 0-100)
last_retry	Number of retries at this rate since last table update
last_success	Number of successful transmissions at this rate since last table update
last_attempts	Number of times this rate was attempted since last table update
total_success	Total number of times this rate had a successful transmission
total_attempts	Total number of times this rate was attempted
mpdu_success	The number of successful MPDUs in acknowledged A-MPDUs at this rate since last table update
mpdu_failures	The number of failed MPDUs in acknowledged A-MPDUs at this rate since last table update
mac_address	The mac address of the peer that this rate applies to

Table 4 - MMRC debugfs Field Descriptions

6 Revision History

Release Number	Release Date	Release Notes
01	08/08/2024	Initial release.
02	3/06/2025	revised MMRC description



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

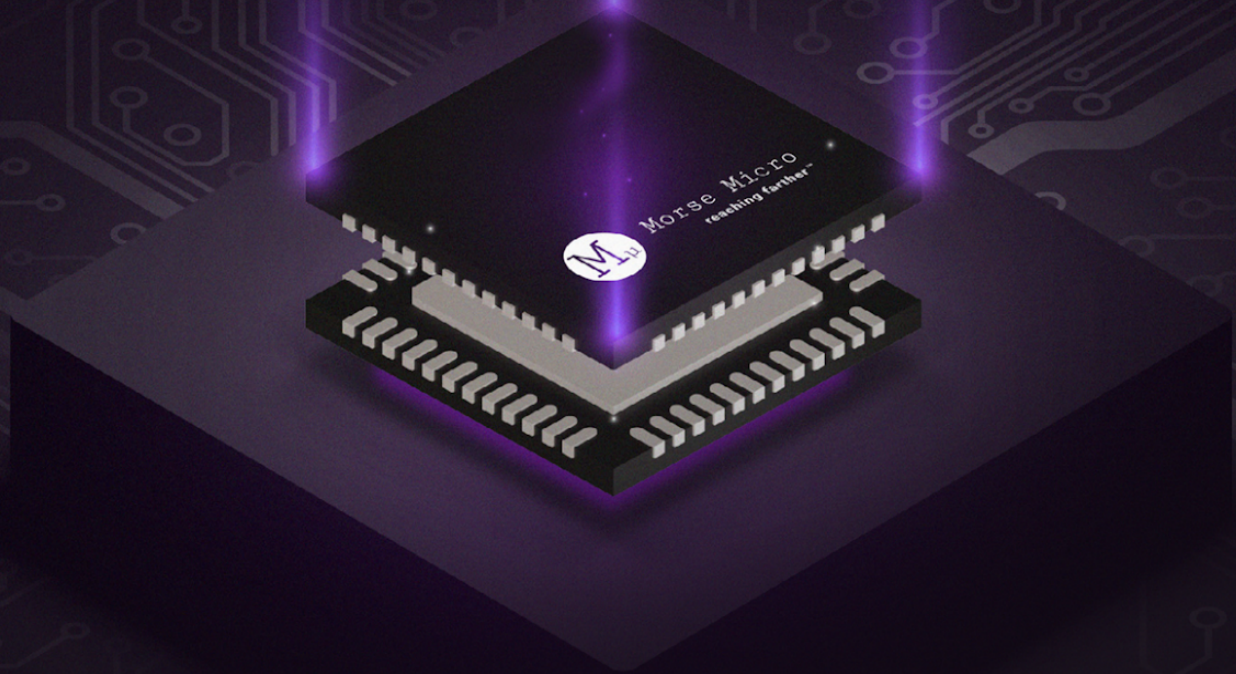
www.morsemicro.com

© 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



Morse Micro
reaching farther™