



Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

WLAN Packet Queue Statistics (PKTQ_STATS)

Associated Part Family: CYW43XX

This application note describes Packet Queue Statistics (PKTQ_STATS), which measures the number of incoming WLAN packets and whether they succeed or fail. It supplements the statistical counters provided by the Cypress WLAN driver to provide a high-level overview of how the system is behaving. This document is intended for engineers and developers who are working with the CYW43XX chipsets.

1 Introduction

Packet Queue Statistics (PKTQ_STATS) helps developers understand the flow of transmit packets through the queues of the Cypress 43xx chip sets. It provides a way to observe the effect of problems such as dropped packets or queue congestion, and helps understand system behavior when packet loss or poor throughput happens. Statistics are collected by precedence type.

The feature is best used by dynamically making debug information requests on a regular, periodic basis to observe behavior changes over time. In particular, physical layer problems are transient and unpredictable, and their consequences are on time scales of few seconds.

The statistics are link-specific: there is a separate debug trace for each Wi-Fi peer, referenced by the MAC address. Further distinction is made between the different transmit modules (aggregated traffic - AMPDU, power save). There are also statistics available at the system level to observe the transmit common queue.

Packet Queue Statistics (PKTQ_STATS) counts the number of incoming WLAN packets and whether they succeed or fail. It supplements the statistical counters provided by the Cypress WLAN driver to provide a high-level overview of how the system is behaving.

PKTQ_STATS is a Cypress WLAN proprietary command that is mapped to an IOVAR (IOV_PKTQ_STATS) where it dumps packet queue log information for the Common [C], AMPDU [A], NAR [N], or power-save queues [P].

A:, N:, or P: are used to prefix a MAC address (a colon: separator is necessary), or else C: is used alone.

```
# wl pktq_stats
```

Common queue:

prec:	rqstd,	stored,	dropped,	retried,	rtsfail,	rtrydrop,	psretry,	acked,	utlisatn,	q length,	Data Mbits
00:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
01:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
02:	0,	0,	0,	3,	0,	0,	0,	3,	0,	1368,	0.00
03:	3,	3,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
04:	208779,	208779,	0,	9653,	764,	27,	0,	81548,	0,	1368,	0.05
05:	2766,	2766,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
06:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
07:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
08:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
09:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
10:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
11:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
12:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
13:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
14:	0,	0,	0,	97,	0,	0,	0,	96,	0,	1368,	0.00

The usefulness and power of `pktq_stats` is increased when data is logged repeatedly in real-time.

The `pktq_stats` counters are self-zeroing: they always self-clear, which means that each new set of data is incremental to the previous.

Packet Queue Statistics is enabled using the build switch `PKTQ_LOG`. It increases memory usage and so may be disabled on dongle builds. For router builds it is normally enabled. It is available on external software releases when `PKTQ_LOG` is set.

1.1 Cypress Part Numbering Scheme

Cypress is converting the acquired IoT part numbers from Broadcom to the Cypress part numbering scheme. Due to this conversion, there is no change in form, fit, or function as a result of offering the device with Cypress part number marking. The table provides Cypress ordering part number that matches an existing IoT part number

Table 1. Mapping Table for Part Number between Broadcom and Cypress

Broadcom Part Number	Cypress Part Number
BCM43XX	CYW43XX

2 IoT Resources

Cypress provides a wealth of data at <http://www.cypress.com/internet-things-iot> to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (<http://community.cypress.com/>).

2.1 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

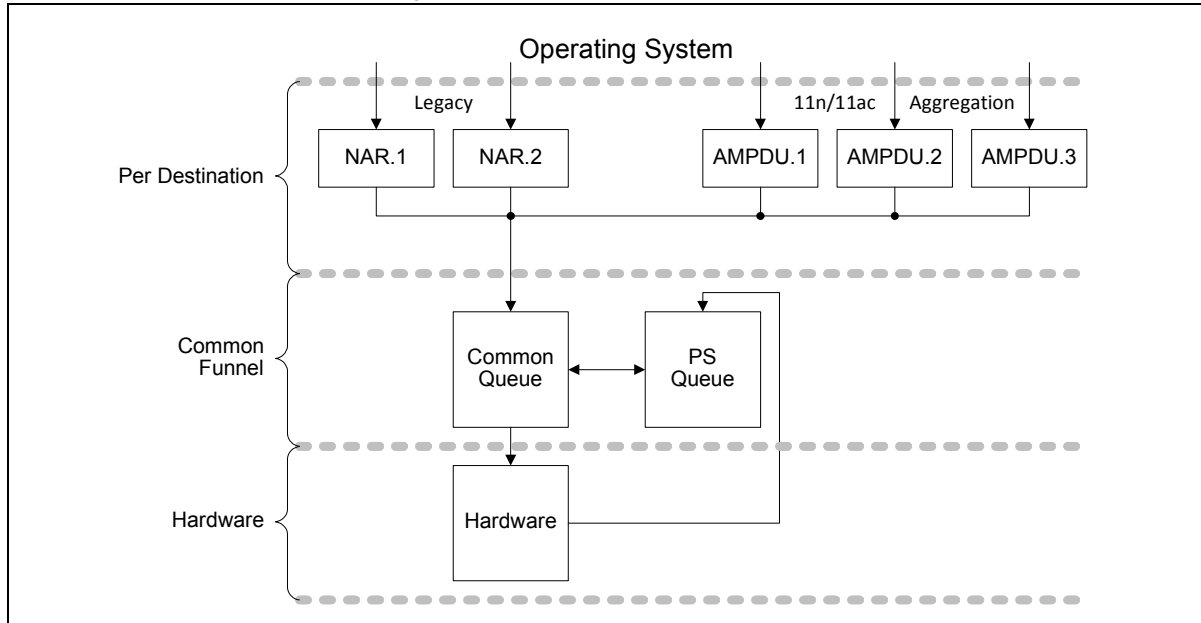
For a comprehensive list of acronyms and other terms used in Cypress documents, go to <http://www.cypress.com/glossary>.

3 Queues Overview

There are four queues:

- Common:
 - Non-aggregated traffic:
 - Not using NAR, traffic is queued here immediately
 - Using NAR, traffic is queued here after being released from the NAR module.
All signaling/control, true multicast, and aggregated traffic are queued here after they are released.
- AMPDU: aggregated traffic prior to being released to the common queue.
- PS: traffic held for destination in power save mode.
- NAR (Non-aggregated regulated): optional counterpart to AMPDU so that non-aggregated traffic does not have an advantage over aggregated traffic (to balance the length of the data path).

Figure 1. Queue Overview



3.1 Queue Precedence

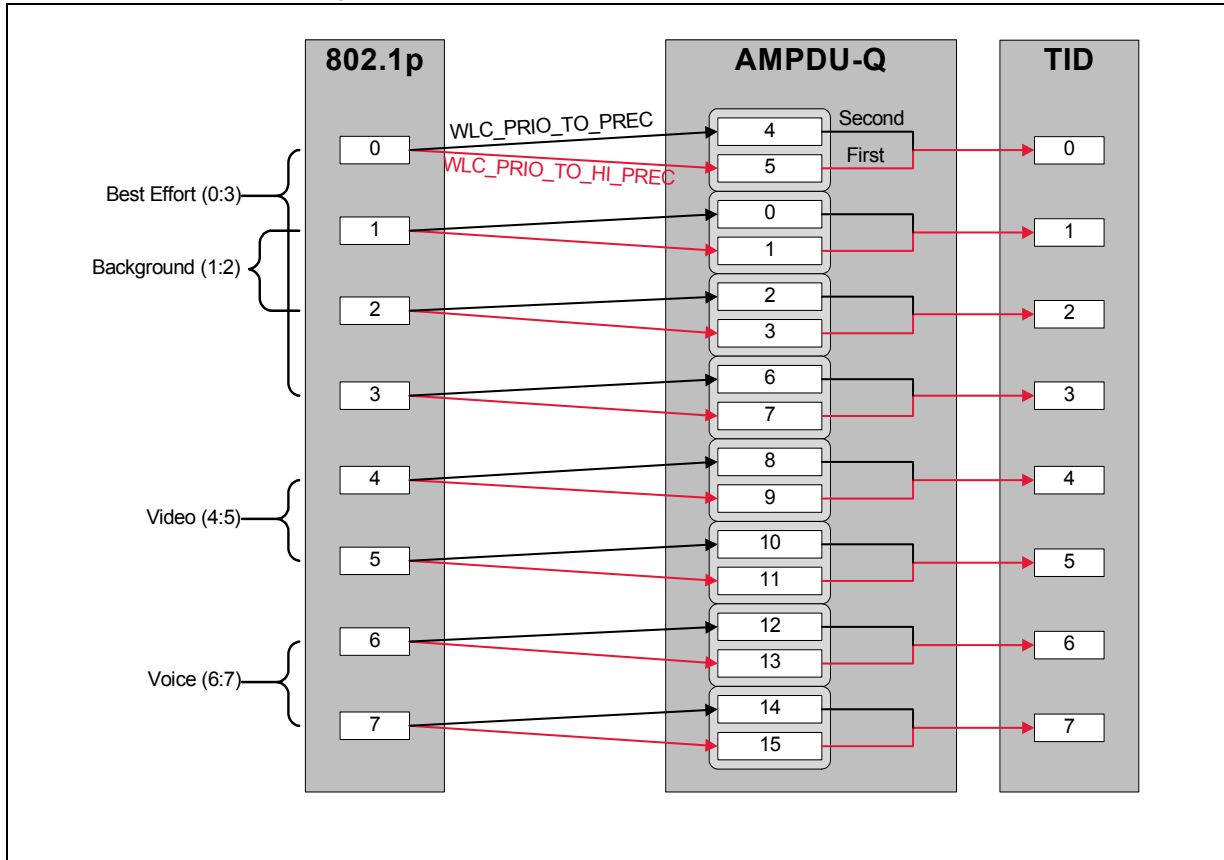
There are 16 levels of precedence (0 to 15):

- Background = 0~3
- Best Effort = 4~7
- Video = 8~11
- Voice = 12~15

Figure 2. Queue Precedence

prec:	rqstd,	stored,	dropped,	retried,	rtsfail,	rtrydrop,	psretry,	acked,	utilisatn,	q length,	Data M/bits/s
00:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
01:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
02:	0,	0,	0,	0,	0,	0,	0,	3,	0,	1368,	0.00
03:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
Background											
04:	208779,	208779,	0,	9653,	764,	27,	0,	81458,	0,	1368,	0.05
05:	2766,	2766,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
06:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
07:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
Best Effort											
08:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
09:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
10:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
11:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
Video											
12:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
13:	0,	0,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
14:	0,	0,	0,	97,	0,	0,	0,	96,	0,	1368,	0.00
15:	96,	96,	0,	0,	0,	0,	0,	0,	0,	1368,	0.00
Voice											

Figure 3. Overview of Queues – Precedence



Even precedence is normal; odd (+1) precedence is the equivalent TID, but has a higher precedence. The higher precedence is used for packet retries, exit from power-save mode, and favored traffic (AMPDU only).

Normal, non-specific traffic is often the traffic of interest, so Best Effort (precedence 4) is the precedence most frequently observed.

4 WL PKTQ_STATS

4.1 Introduction

When used with the Cypress 43xx WL utility, `wl pktq_stats` dumps packet queue log information for Common, AMPDU, NAR, or Power-save queues.

The + option after the colon gives more details. Up to four parameters may be given. The common queue is default when no parameters are supplied Use `/<PREC>` as suffix to restrict to certain prec indices; multiple `/<PREC>/<PREC>/...` can be used.

When `//` or `C://` is used as a suffix to the MAC address automatic logging is enabled for all precedence as they are seen. Full automatic operation is also possible with the shorthand A: (or A://), P: (or P://) etc., which scans through all known addresses for those parameters that take a MAC address.

```
wl pktq_stats [C:[+]]|[A:[+]|P:[+]|N:[+]<xx:xx:xx:xx:xx:xx>[/<PREC>/<PREC>]][///
]... },
```

```
wl pktq_stats [C:[+]]|[A:[+]|P:[+]<xx:xx:xx:xx:xx:xx>]...
```

Note: NAR is also an option (if enabled) using `N:[+]xx:xx:xx:xx:xx:xx`

4.2 Examples

The commands below are equivalent and dump the common queue:

```
wl pktq_stats
```

```
wl pktq_stats c:
```

```
# wl pktq_stats
```

Common queue:														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	qlength	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
04:	1306	1306	0	25	158	0	0	633	5	200	0.00	136.67	0.0	2.0
05:	42	42	0	-	-	0	0	0	0	200	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
08:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
09:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
10:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
11:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
12:	24	24	0	2	10	0	0	26	1	200	0.00	124.85	0.0	3.2
13:	2	2	0	-	-	0	0	0	0	200	0.00	-	-	-
14:	0	0	0	2169	0	106	0	526	0	200	0.00	0.00	0.1	0.0
15:	632	632	0	-	-	0	0	0	0	200	0.00	-	-	-

For MAC address 00:11:22:33:44:55, get the AMPDU queue.

```
# wl pktq_stats a:00:11:22:33:44:55
```

AMPDU queue 00:11:22:33:44:55														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	588	588	0	4	34	0	0	588	2	1024	0.00	143.66	0.0	1.7
01:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-

Get the AMPDU queue, the PS queue, and the Common queue all at once.

```
wl pktq_stats a: 00:11:22:33:44:55 p: 00:11:22:33:44:55 c:
```

AMPDU queue 00:11:22:33:44:55														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-

Power save queue 00:11:22:33:44:55														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)

Common Queue:														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
04:	1558	1558	0	46	29	0	0	629	2	200	0.00	130.41	0.0	2.1
05:	54	54	0	-	-	0	0	0	0	200	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
08:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
09:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
10:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
11:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
12:	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.0	0.0
13:	0	0	0	-	-	0	0	0	0	200	0.00	-	-	-
14:	0	0	0	3757	0	172	0	999	0	200	0.01	0.00	1.0	0.0
15:	1171	1171	0	-	-	0	0	0	0	200	0.00	-	-	-

AMPDU queue 00:11:22:33:44:55

prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-

Power save queue 00:11:22:33:44:55

prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
08:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
09:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
10:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
11:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
12:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
13:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-
14:	0	0	0	0	0	0	0	0	0	512	0.00	0.00	0.0	0.0
15:	0	0	0	-	-	0	0	0	0	512	0.00	-	-	-

For two different STAs, get the AMPDU queue information

```
#wl pktq_stats a:00:11:22:33:44:55 a:8C:70:5A:2A:84:F4
```

AMPDU queue 00:11:22:33:44:55														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
01:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-

AMPDU queue 8c:70:5a:2a:84:f4														
prec:	rqstd	stored	dropped	retried	rtsfail	rtrydrop	psretry	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	%air	%effcy (v5)
00:	1	1	0	0	0	0	0	1	0	1024	0.00	144.00	0.0	2.1
01:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
02:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
03:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
04:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
05:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-
06:	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.0	0.0
07:	0	0	0	-	-	0	0	0	0	1024	0.00	-	-	-

Get the AMPDU and Common queues with extra information

```
# wl pktq_stats c:+ a:+ 00:11:22:33:44:55
```

Common queue:

prec:	rqstd	stored	selfsave	saved	fulldrop	dropped	sacrficd	retried	rtsfail	rtrydrop	psretry	supprssd	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	Rate Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
01:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
02:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
03:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
04:	73	73	0	0	0	0	0	3	0	0	0	0	14	0	200	0.00	102.86	84.71	0.0	2.0
05:	4	4	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
06:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
07:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
08:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
09:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
10:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
11:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
12:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0.00	0.00	0.00	0.0	0.0
13:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-
14:	0	0	0	0	0	0	0	233	0	15	0	0	54	0	200	0.00	0.00	0.00	1.0	0.0
15:	69	69	0	0	0	0	0	-	-	0	0	0	0	0	200	0.00	-	-	-	-

AMPDU queue 00:11:22:33:44:55

prec:	rqstd	stored	selfsave	saved	fulldrop	dropped	sacrficd	retried	rtsfail	rtrydrop	psretry	supprssd	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	Rate Mbits/s	%air	%effcy (v5)
00:	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1024	0.00	144.00	144.00	0.0	1.5
01:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
02:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
03:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
04:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
05:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
06:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
07:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-

AMPDU queue 00:11:22:33:44:55

prec:	rqstd	stored	selfsave	saved	fulldrop	dropped	sacrficd	retried	rtsfail	rtrydrop	psretry	supprssd	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	Rate Mbits/s	%air	%effcy (v5)
00:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
01:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
02:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
03:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
04:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
05:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
06:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
07:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-

AMPDU queue 8c:70:5a:2a:84:f4

prec:	rqstd	stored	selfsave	saved	fulldrop	dropped	sacrficd	retried	rtsfail	rtrydrop	psretry	supprssd	acked	utlisatn	q length	Data Mbits/s	Phy Mbits/s	Rate Mbits/s	%air	%effcy (v5)
00:	8	8	0	0	0	0	0	0	0	0	0	0	8	0	1024	0.00	144.00	144.00	0.0	1.8
01:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
02:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
03:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
04:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
05:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-
06:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1024	0.00	0.00	0.00	0.0	0.0
07:	0	0	0	0	0	0	0	-	-	0	0	0	0	0	1024	0.00	-	-	-	-

5 Output Descriptions (Includes + option)

- prec is the precedence, as explained in LINK.
- rqstd is the number of packets sent by the operating system to the driver.
If this is low, then the throughput is necessarily low because the driver can only send what it has been given. On the other hand, more packets may be sent than it is possible to accept because the physical link is not fast enough.
- stored shows the actual number of packets accepted by the driver, which may be less than the number requested if the link is too slow and the buffers fill up.
- Some counters are not shown in the concise output format. They describe how packets are dropped or saved because of packet priority:
 - Selfsave is the number packets saved because an older packet from the same queue has been dropped.
 - Saved is the number of packets saved because a lowest priority queue has given away one packet.
 - Fulldrop is the number of packets dropped because the packet queue is full with higher precedence packets.
 - Sacrificed is the number of packets dropped in order to save one from a higher priority queue
- dropped records the number of excess packets that were discarded.
- retried is the number of times a packet was sent if this is more than once. For example, a packet that is sent four times with three failures and eventual success, will record a retry count of 3. If all packets are sent successfully the first time the retry count will be 0. A non-zero value indicates transmission problems.
- rtsfail is the number of times RTS was sent without receiving a CTS response. This should be zero: a non-zero value indicates a problem.
- retrydrop records the loss of a packet because the retry count exceeded the retry count limit. Often this corresponds to a change in dynamic behavior of throughput. TCP, for example, will show a drop off when there is a packet drop shown here.
- psretry is the number of times the Pspretend was activated.
- supprsd is number of packets that were supposed to be transmitted, but that were rejected by the micro-code. For example, when STA enters the power-save mode, packets are suppressed and queued to the power save queue.
- acked is the number of packets that were successfully sent. Even if a packet is retried many times, if it eventually succeeds it will be measured here. *This count is only supported for aggregated AMPDU traffic.*
- utlisatn is the maximum number of packets queued at once.
- q length is the maximum queue capacity.
- Data Mbits/s is the data throughput (average since previous).
- Phy Mbits/s is the PHY rate (average since previous).
- Rate Mbits/s is average primary PHY rate for all packets, regardless of whether they were ACKd. The primary PHY rate is the first modulation used for the first transmission attempt.
- %air is the percentage of air occupied by the WLAN traffic.
- %effcy is the average efficiency of the transmission.

Note:

- The counters are automatically cleared each and every time they are requested (using the command `pktq_stats`). It is found that this way the change in behavior with time is easier to observe, rather than maintain a cumulative count which must be post-processed to obtain deltas. One relevant operating point is that if `wl pktq_stats` has not been used recently, then the counters will contain historical data for previous conditions that is unlikely to be relevant. Hence, the first time the command is issued, the values should be disregarded, but from that point on further requests will show the incremental changes that are needed to investigate the issue.
- As a consequence of data auto-clearing of, you may get unpredictable results if you make requests for data by running the `wl pktq_stats` concurrently in multiple sessions, particularly if you are requesting information about the same queue in the parallel session. Therefore it is recommended to only request data for any particular queue once across all sessions in use.
- Data is presented in tabular form. Because of the number of data columns, there is a choice between concise and full output. For the concise format, some of the data columns are not shown, reducing the required screen width to view the tabulated output. By default the output is the concise form; the full form is selected by additional command line parameter.
- Generally the concise format is most often used and shows statistics for investigating packet loss when there are no competing traffic streams. In case where packets are dropped because of competition between multiple streams, the full format output would be needed.
- There is no clear command for `pktq_stats`.
- Every time you ask for `pktq_stats`, the counters are zeroed automatically. Only the `pktq_stats` you asked for are cleared the `pktq_stats` for queues you did not specify continue to accumulate.
 - The first time you do `pktq_stats`, the counters contain historical information. Prior to running some test, you should therefore do a `pktq_stats` command to clear/reset. Then do `pktq_stats` throughout your test or at the end of it.
 - If you have multiple sessions, for example, concurrent Telnet to the router, be careful about running the same `pktq_stats` at once. Because the data is cleared, the data will not be consistent It is very useful to see `pktq_stats` on repetitive basis. You can reduce the amount of info shown by using GREP

6 Timestamp

For software debugging, a timestamp can be added.

6.1 Timestamp Use

```
# wl msglevel +time
# while sleep 1; do wl pktq_stats a:00:90:4c:09:38:85 a:00:10:18:A9:10:2E | grep -e 04: -e queue -e prec;;
echo -----; done

[32:43.93639]:AMPDU queue 00:90:4c:09:38:85
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  13102,  13102,  0,  1,  46,  0,  0,  13052,  84,  1024,  67.46
[32:43.93639]:AMPDU queue 00:10:18:a9:10:2e
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  11681,  11681,  0,  14,  53,  0,  0,  11556,  94,  1024,  59.59
-----
[32:45.07078]:AMPDU queue 00:90:4c:09:38:85
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  6428,  6428,  0,  2,  28,  0,  0,  6440,  97,  1024,  64.90
[32:45.07078]:AMPDU queue 00:10:18:a9:10:2e
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  6111,  6111,  0,  11,  36,  0,  0,  6092,  83,  1024,  61.08
-----
[32:46.22835]:AMPDU queue 00:90:4c:09:38:85
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  7362,  7362,  0,  0,  19,  0,  0,  7351,  91,  1024,  72.53
[32:46.22835]:AMPDU queue 00:10:18:a9:10:2e
prec:  rqstd,  stored,  dropped,  retried,  rtsfail,rtrydrop,  psretry,  acked,  utlisatn,q length,  Mbits/s
04:  5930,  5930,  0,  12,  21,  0,  0,  6010,  95,  1024,  59.12
-----
```

Document History

Document Title: AN214851 - WLAN Packet Queue Statistics (PKTQ_STATS)				
Document Number: 002-14851				
Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
**	-	-	05/05/15	43XX-AN2000-R: Initial release
*A	5529739	UTSV	11/23/2016	Updated to Cypress template
*B	5824495	AESATMP8	07/19/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Video](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



© Cypress Semiconductor Corporation, 2015-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1s) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.