# Marvell(Aquantia) AQtion Driver

For the aQuantia Multi-Gigabit PCI Express Family of Ethernet Adapters

## Identifying Your Adapter

The driver in this release is compatible with AQC-100, AQC-107, AQC-108 based ethernet adapters.

### SFP+ Devices (for AQC-100 based adapters)

This release tested with passive Direct Attach Cables (DAC) and SFP+/LC Optical Transceiver.

## Configuration

### Viewing Link Messages

Link messages will not be displayed to the console if the distribution is restricting system messages. In order to see network driver link messages on your console, set dmesg to eight by entering the following:

```
dmesg -n 8
```

> **Note**
> This setting is not saved across reboots.

### Jumbo Frames

The driver supports Jumbo Frames for all adapters. Jumbo Frames support is enabled by changing the MTU to a value larger than the default of 1500. The maximum value for the MTU is 16000. Use the *ip* command to increase the MTU size. For example:

```
ip link set mtu 16000 dev enp1s0
```

### ethtool

The driver utilizes the ethtool interface for driver configuration and diagnostics, as well as displaying statistical information. The latest ethtool version is required for this functionality.

### NAPI

NAPI (Rx polling mode) is supported in the atlantic driver.

## Supported ethtool options

### Viewing adapter settings

```
ethtool <ethX>
```

Output example:

```
Settings for enp1s0:
  Supported ports: [ TP ]
  Supported link modes:   100baseT/Full
                          1000baseT/Full
                          10000baseT/Full
                          2500baseT/Full
                          5000baseT/Full
  Supported pause frame use: Symmetric
  Supports auto-negotiation: Yes
  Supported FEC modes: Not reported
  Advertised link modes:  100baseT/Full
                          1000baseT/Full
                          10000baseT/Full
                          2500baseT/Full
                          5000baseT/Full
  Advertised pause frame use: Symmetric
  Advertised auto-negotiation: Yes
  Advertised FEC modes: Not reported
  Speed: 10000Mb/s
  Duplex: Full
  Port: Twisted Pair
  PHYAD: 0
  Transceiver: internal
  Auto-negotiation: on
  MDI-X: Unknown
  Supports Wake-on: g
  Wake-on: d
```

```
Link detected: yes
```

> **Note**
>
> AQrate speeds (2.5/5 Gb/s) will be displayed only with linux kernels > 4.10. But you can still use these speeds:
>
> ```
> ethtool -s eth0 autoneg off speed 2500
> ```

## Viewing adapter information

```
ethtool -i <ethX>
```

Output example:

```
driver: atlantic
version: 5.2.0-050200rc5-generic-kern
firmware-version: 3.1.78
expansion-rom-version:
bus-info: 0000:01:00.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

## Viewing Ethernet adapter statistics

```
ethtool -S <ethX>
```

Output example:

```
NIC statistics:
   InPackets: 13238607
   InUCast: 13293852
   InMCast: 52
   InBCast: 3
   InErrors: 0
   OutPackets: 23703019
   OutUCast: 23704941
   OutMCast: 67
   OutBCast: 11
   InUCastOctects: 213182760
   OutUCastOctects: 22698443
   InMCastOctects: 6600
   OutMCastOctects: 8776
   InBCastOctects: 192
   OutBCastOctects: 704
   InOctects: 2131839552
   OutOctects: 226938073
   InPacketsDma: 95532300
   OutPacketsDma: 59503397
   InOctetsDma: 1137102462
   OutOctetsDma: 2394339518
   InDroppedDma: 0
   Queue[0] InPackets: 23567131
   Queue[0] OutPackets: 20070028
   Queue[0] InJumboPackets: 0
   Queue[0] InLroPackets: 0
   Queue[0] InErrors: 0
   Queue[1] InPackets: 45428967
   Queue[1] OutPackets: 11306178
   Queue[1] InJumboPackets: 0
   Queue[1] InLroPackets: 0
   Queue[1] InErrors: 0
   Queue[2] InPackets: 3187011
   Queue[2] OutPackets: 13080381
   Queue[2] InJumboPackets: 0
   Queue[2] InLroPackets: 0
   Queue[2] InErrors: 0
   Queue[3] InPackets: 23349136
   Queue[3] OutPackets: 15046810
   Queue[3] InJumboPackets: 0
   Queue[3] InLroPackets: 0
   Queue[3] InErrors: 0
```

## Interrupt coalescing support

ITR mode, TX/RX coalescing timings could be viewed with:

```
ethtool -c <ethX>
```

and changed with:

```
ethtool -C <ethX> tx-usecs <usecs> rx-usecs <usecs>
```

To disable coalescing:

```
ethtool -C <ethX> tx-usecs 0 rx-usecs 0 tx-max-frames 1 tx-max-frames 1
```

## Wake on LAN support

WOL support by magic packet:

```
ethtool -s <ethX> wol g
```

To disable WOL:

```
ethtool -s <ethX> wol d
```

## Set and check the driver message level

Set message level

```
ethtool -s <ethX> msglvl <level>
```

Level values:

| | |
|---|---|
| 0x0001 | general driver status. |
| 0x0002 | hardware probing. |
| 0x0004 | link state. |
| 0x0008 | periodic status check. |
| 0x0010 | interface being brought down. |
| 0x0020 | interface being brought up. |
| 0x0040 | receive error. |
| 0x0080 | transmit error. |
| 0x0200 | interrupt handling. |
| 0x0400 | transmit completion. |
| 0x0800 | receive completion. |
| 0x1000 | packet contents. |
| 0x2000 | hardware status. |
| 0x4000 | Wake-on-LAN status. |

By default, the level of debugging messages is set 0x0001(general driver status).

Check message level

```
ethtool <ethX> | grep "Current message level"
```

If you want to disable the output of messages:

```
ethtool -s <ethX> msglvl 0
```

## RX flow rules (ntuple filters)

There are separate rules supported, that applies in that order:

1. 16 VLAN ID rules
2. 16 L2 EtherType rules
3. 8 L3/L4 5-Tuple rules

The driver utilizes the ethtool interface for configuring ntuple filters, via `ethtool -N <device> <filter>`.

To enable or disable the RX flow rules:

```
ethtool -K ethX ntuple <on|off>
```

When disabling ntuple filters, all the user programed filters are flushed from the driver cache and hardware. All needed filters must be re-added when ntuple is re-enabled.

Because of the fixed order of the rules, the location of filters is also fixed:

- Locations 0 - 15 for VLAN ID filters
- Locations 16 - 31 for L2 EtherType filters
- Locations 32 - 39 for L3/L4 5-tuple filters (locations 32, 36 for IPv6)

The L3/L4 5-tuple (protocol, source and destination IP address, source and destination TCP/UDP/SCTP port) is compared against 8 filters. For IPv4, up to 8 source and destination addresses can be matched. For IPv6, up to 2 pairs of addresses can be supported. Source and destination ports are only compared for TCP/UDP/SCTP packets.

To add a filter that directs packet to queue 5, use `<-N|-U|--config-nfc|--config-ntuple>` switch:

```
ethtool -N <ethX> flow-type udp4 src-ip 10.0.0.1 dst-ip 10.0.0.2 src-port 2000 dst-port 2001 action 5 <lo
```

- action is the queue number.
- loc is the rule number.

For `flow-type ip4|udp4|tcp4|sctp4|ip6|udp6|tcp6|sctp6` you must set the loc number within 32 - 39. For `flow-type ip4|udp4|tcp4|sctp4|ip6|udp6|tcp6|sctp6` you can set 8 rules for traffic IPv4 or you can set 2 rules for traffic IPv6. Loc number traffic IPv6 is 32 and 36. At the moment you can not use IPv4 and IPv6 filters at the same

time.

Example filter for IPv6 filter traffic:

```
sudo ethtool -N <ethX> flow-type tcp6 src-ip 2001:db8:0:f101::1 dst-ip 2001:db8:0:f101::2 action 1 loc 32
sudo ethtool -N <ethX> flow-type ip6 src-ip 2001:db8:0:f101::2 dst-ip 2001:db8:0:f101::5 action -1 loc 36
```

Example filter for IPv4 filter traffic:

```
sudo ethtool -N <ethX> flow-type udp4 src-ip 10.0.0.4 dst-ip 10.0.0.7 src-port 2000 dst-port 2001 loc 32
sudo ethtool -N <ethX> flow-type tcp4 src-ip 10.0.0.3 dst-ip 10.0.0.9 src-port 2000 dst-port 2001 loc 33
sudo ethtool -N <ethX> flow-type ip4 src-ip 10.0.0.6 dst-ip 10.0.0.4 loc 34
```

If you set action -1, then all traffic corresponding to the filter will be discarded.

The maximum value action is 31.

The VLAN filter (VLAN id) is compared against 16 filters. VLAN id must be accompanied by mask 0xF000. That is to distinguish VLAN filter from L2 Ethertype filter with UserPriority since both User Priority and VLAN ID are passed in the same 'vlan' parameter.

To add a filter that directs packets from VLAN 2001 to queue 5:

```
ethtool -N <ethX> flow-type ip4 vlan 2001 m 0xF000 action 1 loc 0
```

L2 EtherType filters allows filter packet by EtherType field or both EtherType and User Priority (PCP) field of 802.1Q. UserPriority (vlan) parameter must be accompanied by mask 0x1FFF. That is to distinguish VLAN filter from L2 Ethertype filter with UserPriority since both User Priority and VLAN ID are passed in the same 'vlan' parameter.

To add a filter that directs IP4 packess of priority 3 to queue 3:

```
ethtool -N <ethX> flow-type ether proto 0x800 vlan 0x600 m 0x1FFF action 3 loc 16
```

To see the list of filters currently present:

```
ethtool <-u|-n|--show-nfc|--show-ntuple> <ethX>
```

Rules may be deleted from the table itself. This is done using:

```
sudo ethtool <-N|-U|--config-nfc|--config-ntuple> <ethX> delete <loc>
```

- loc is the rule number to be deleted.

Rx filters is an interface to load the filter table that funnels all flow into queue 0 unless an alternative queue is specified using "action". In that case, any flow that matches the filter criteria will be directed to the appropriate queue. RX filters is supported on all kernels 2.6.30 and later.

## RSS for UDP

Currently, NIC does not support RSS for fragmented IP packets, which leads to incorrect working of RSS for fragmented UDP traffic. To disable RSS for UDP the RX Flow L3/L4 rule may be used.

Example:

```
ethtool -N eth0 flow-type udp4 action 0 loc 32
```

## UDP GSO hardware offload

UDP GSO allows to boost UDP tx rates by offloading UDP headers allocation into hardware. A special userspace socket option is required for this, could be validated with /kernel/tools/testing/selftests/net/:

```
udpgso_bench_tx -u -4 -D 10.0.1.1 -s 6300 -S 100
```

Will cause sending out of 100 byte sized UDP packets formed from single 6300 bytes user buffer.

UDP GSO is configured by:

```
ethtool -K eth0 tx-udp-segmentation on
```

## Private flags (testing)

Atlantic driver supports private flags for hardware custom features:

```
$ ethtool --show-priv-flags ethX

Private flags for ethX:
DMASystemLoopback  : off
PKTSystemLoopback  : off
DMANetworkLoopback : off
PHYInternalLoopback: off
PHYExternalLoopback: off
```

Example:

```
$ ethtool --set-priv-flags ethX DMASystemLoopback on
```

DMASystemLoopback: DMA Host loopback. PKTSystemLoopback: Packet buffer host loopback. DMANetworkLoopback: Network side loopback on DMA block. PHYInternalLoopback: Internal loopback on Phy.

PHYExternalLoopback: External loopback on Phy (with loopback ethernet cable).

## Command Line Parameters

The following command line parameters are available on atlantic driver:

### aq_itr -Interrupt throttling mode

Accepted values: 0, 1, 0xFFFF

Default value: 0xFFFF

| | |
|---|---|
| 0 | Disable interrupt throttling. |
| 1 | Enable interrupt throttling and use specified tx and rx rates. |
| 0xFFFF | Auto throttling mode. Driver will choose the best RX and TX interrupt throtting settings based on link speed. |

### aq_itr_tx - TX interrupt throttle rate

Accepted values: 0 - 0x1FF

Default value: 0

TX side throttling in microseconds. Adapter will setup maximum interrupt delay to this value. Minimum interrupt delay will be a half of this value

### aq_itr_rx - RX interrupt throttle rate

Accepted values: 0 - 0x1FF

Default value: 0

RX side throttling in microseconds. Adapter will setup maximum interrupt delay to this value. Minimum interrupt delay will be a half of this value

> **Note**
>
> ITR settings could be changed in runtime by ethtool -c means (see below)

## Config file parameters

For some fine tuning and performance optimizations, some parameters can be changed in the {source_dir}/aq_cfg.h file.

### AQ_CFG_RX_PAGEORDER

Default value: 0

RX page order override. Thats a power of 2 number of RX pages allocated for each descriptor. Received descriptor size is still limited by AQ_CFG_RX_FRAME_MAX.

Increasing pageorder makes page reuse better (actual on iommu enabled systems).

### AQ_CFG_RX_REFILL_THRES

Default value: 32

RX refill threshold. RX path will not refill freed descriptors until the specified number of free descriptors is observed. Larger values may help better page reuse but may lead to packet drops as well.

### AQ_CFG_VECS_DEF

Number of queues

Valid Range: 0 - 8 (up to AQ_CFG_VECS_MAX)

Default value: 8

Notice this value will be capped by the number of cores available on the system.

### AQ_CFG_IS_RSS_DEF

Enable/disable Receive Side Scaling

This feature allows the adapter to distribute receive processing across multiple CPU-cores and to prevent from overloading a single CPU core.

Valid values

| | |
|---|---|
| 0 | disabled |
| 1 | enabled |

Default value: 1

### AQ_CFG_NUM_RSS_QUEUES_DEF

Number of queues for Receive Side Scaling

Valid Range: 0 - 8 (up to AQ_CFG_VECS_DEF)

Default value: AQ_CFG_VECS_DEF

### AQ_CFG_IS_LRO_DEF

Enable/disable Large Receive Offload

This offload enables the adapter to coalesce multiple TCP segments and indicate them as a single coalesced unit to the OS networking subsystem.

The system consumes less energy but it also introduces more latency in packets processing.

Valid values

| | |
|---|---|
| 0 | disabled |
| 1 | enabled |

Default value: 1

### AQ_CFG_TX_CLEAN_BUDGET

Maximum descriptors to cleanup on TX at once.

Default value: 256

After the aq_cfg.h file changed the driver must be rebuilt to take effect.

## Support

If an issue is identified with the released source code on the supported kernel with a supported adapter, email the specific information related to the issue to aqn_support@marvell.com

## License

aQuantia Corporation Network Driver