

## How can I monitor the fullness of the receive ring in Sniffer10G?

### Model:

ARC Series C Adapters (10G-PCIE2-8C2-2S)

### Software:

Sniffer10G

### Operating System:

Supports both Windows and Linux operating systems.

### Information:

For full details, please read the **Sniffer10G API Documentation**, available in the **/share/doc/directory** in the software distribution.

Sniffer10G version 2.x/3.x has this function:

```
int snf_ring_recv_qinfo(snf_ring_t ring, struct snf_ring_qinfo *)'
```

With the structure as defined below:

```
/** Queue consumption information */  
  
struct snf_ring_qinfo {  
    uintptr_t q_avail; /**< Amount of data available not yet received  
(approximate) */  
    uintptr_t q_borrowed; /**< Amount of data currently borrowed (exact) */  
    uintptr_t q_free; /**< Amount of free space still available (approximate)  
*/  
};
```

All these values are in bytes.

Sniffer10G also has this function to obtain statistics from a receive ring.

```
int snf_ring_getstats(snf_ring_t ringh, struct snf_ring_stats *stats);
```

With the structure as defined below:

```
/**  
 * Structure to return statistics from a ring. The Hardware-specific  
 * counters apply to all rings as they are counted before any  
 * demultiplexing to a ring is applied.  
 */
```



```

struct snf_ring_stats {
    uint64_t  nic_pkt_rcv;           /** Number of packets received by Hardware
Interface */
    uint64_t  nic_pkt_overflow;     /** Number of packets dropped by Hardware
Interface */
    uint64_t  nic_pkt_bad;          /** Number of Bad CRC/PHY packets seen by
Hardware Interface */
    uint64_t  ring_pkt_rcv;         /** Number of packets received into the
receive ring */
    uint64_t  ring_pkt_overflow;    /** Number of packets dropped because of
insufficient space in receive ring */
    uint64_t  nic_bytes_rcv;        /** Number of raw bytes received by the
Hardware Interface on
                                     all rings. Each Ethernet data packet
                                     includes 8 bytes of HW
                                     header, 4 bytes of CRC and the result is
                                     aligned to 16 bytes
                                     such that a minimum size 60 byte packet
                                     counts for 80 bytes. */
    uint64_t  snf_pkt_overflow;     /** Number of packets dropped because of
insufficient space in shared
                                     SNF buffering */
};

```

New packets are dropped if the ring is full. There is no warning for when the ring might overflow. The **snf\_ring\_rcv\_qinfo()** and **snf\_ring\_getstats()** are written to non-invasively check the ring stats and it is left to the application to use these functions to monitor the ring.

<u>Draft</u>	<u>Date</u>	<u>Change</u>
1	7/11/16	Initial Draft
2	7/28/2016	Feedback