NAME

mri_poll - Poll a ring for network data

SYNOPSIS

#include <sys/mac_provider.h>

*mblk_t **
prefix_ring_poll(*void *driver, int poll_bytes*);

INTERFACE LEVEL

Evolving - This interface is still evolving. API and ABI stability is not guaranteed.

PARAMETERS

driver A pointer to the ring's private data that was passed in via the *mri_driver* member of the mac_ring_info(9S) structure as part of the mr_rget(9E) entry point.

poll_bytes The maximum number of bytes that the driver should poll in a given call.

DESCRIPTION

The **mri_poll**() entry point is called by the MAC framework when it wishes to have the driver check the ring specified by *driver* for available data.

The device driver should perform the same logic that it would when it's procesing an interrupt and as described in the *Receiving Data* section of mac(9E). The main difference is that instead of calling mac_ring_rx(9E), it should instead return that data. Also, while an interrupt may map to more than one ring, the driver should only process the ring indicated by *driver*.

Drivers should exercise caution with the locking between the polling, interrupt disabling routines, and the interrupt handler. This lock is generally scoped to a receive ring and is used to synchronize the act of transitioning between polling and handling interrupts. That means that in addition to the **mri_poll**() entry point, the mi_enable(9E) and mi_disable(9E) entry points should synchronize on the same lock when transitioning the card. While the driver does not need to hold this lock across the entire interrupt handler, it should hold it when it is processing and trying to read frames for the specified ring in its interrupt handler.

The driver should limit the number of frames it collects based on the size value present in the *poll_bytes* argument.

RETURN VALUES

Upon successful completion, the device driver should return a message block chain of collected frames.

If no frames are available, then it should return NULL.

SEE ALSO

mac(9E), mac_capab_rings(9E), mac_ring_rx(9E), mi_disable(9E), mi_enable(9E), mr_rget(9E), mac_ring_info(9S)