NAME

lo - software loopback network interface

SYNOPSIS

device loop

DESCRIPTION

The **loop** interface is a software loopback mechanism which may be used for performance analysis, software testing, and/or local communication. As with other network interfaces, the loopback interface must have network addresses assigned for each address family with which it is to be used. These addresses may be set with the appropriate ioctl(2) commands for corresponding address families. The loopback interface should be the last interface configured, as protocols may use the order of configuration as an indication of priority. The loopback should *never* be configured first unless no hardware interfaces exist.

If the transmit checksum offload capability flag is enabled on a loopback interface, checksums will not be generated by IP, UDP, or TCP for packets sent on the interface.

If the receive checksum offload capability flag is enabled on a loopback interface, checksums will not be validated by IP, UDP, or TCP for packets received on the interface.

By default, both receive and transmit checksum flags will be enabled, in order to avoid the overhead of checksumming for local communication where data corruption is unlikely. If transmit checksum generation is disabled, then validation should also be disabled in order to avoid packets being dropped due to invalid checksums.

DIAGNOSTICS

lo%d: can’t handle af%d. The interface was handed a message with addresses formatted in an unsuitable address family; the packet was dropped.

SEE ALSO

inet(4), intro(4)

HISTORY

The **lo** device appeared in 4.2BSD. The current checksum generation and validation avoidance policy appeared in FreeBSD 8.0.