allocateNetbuf

finishInitialization
outputPacket:address:
performCommand:data:

free
initFromDeviceDescription:
attachToNetworkWithAddress:

Transmitting packets transmit:

Setting and handling hardware timeouts

setRelativeTimeout:
relativeTimeout
clearTimeout

Setting and getting the state of the hardware

setRunning:
isRunning
resetAndEnable:

(IONetwork *)attachToNetworkWithAddress:(token_addr_t)address

Invokes registerDevice, sets the node address to address, creates an IONetwork instance, and attac
subsystem by sending the IONetwork an initForNetworkDevice:... message. Besides starting up th
the device, this method also starts up an 802.2-compliant Null SAP interface. Finally, this method
the node address. Returns the IONetwork instance just created.

To determine the value to specify for address, first invoke nodeAddress. If nodeAddress returns a
value. Otherwise, use the hardware's burnt-in address.

You invoke this method at the end of your implementation of initFromDeviceDescription:. You m
resetAndEnable:NO before invoking this method, as described under initFromDeviceDescription:,
specification.

(void)clearTimeout

If a transmission timeout is scheduled, unschedules the timeout. This method is normally invoked
implementation of interruptOccurred.

setRelativeTimeout:, relativeTimeout

(BOOL)earlyTokenEnabled

Returns YES if Early Token Release (ETR) is supported by the station otherwise, returns NO. Stat
can co-exist with non-ETR stations in the ring. The value returned by this method is set by initFro

free

Frees the IOTokenRing instance and its resources and returns nil.

initFromDeviceDescription:(IODeviceDescription *)devDesc

Invokes the superclass implementation, starts an I/O thread (using startIOThread), and sets the dev
unit.

Next, it examines the device configuration table for such parameters as ring speed and early token
the maximum packet size, based on the ring speed. If the ring speed is 4 megabits per second, the
is MAC_INFO_4MB. If the ring speed is 16, the maximum info field size is MAC_INFO_16MB.
size is the maximum info field size plus MAC_HDR_MAX.) These constants are defined in the he
tokendefs.h.

Subclasses of IOTokenRing should implement this method so that it invokes the superclass versio
initFromDeviceDescription:, makes sure the configuration is correct, invokes setMaxInfoFieldSize
device-specific software and hardware initialization, and invokes attachToNetworkWithAddress:.

(BOOL)isRunning

Returns YES if the hardware is currently inserted in the ring otherwise, returns NO.

setRunning:

(unsigned int)maxInfoFieldSize

Returns the maximum size of the info field. This value is used by allocateNetbuf. It's also used as
unit specified to the network subsystem.

setMaxInfoFieldSize:

(token_addr_t)nodeAddress

Returns the node address for this station. Currently, only burnt-in addresses are supported. In the fu
IOTokenRing will be able to initialize the node address from the device configuration table. The va
method is set by attachToNetworkWithAddress:.

(unsigned int)relativeTimeout

Returns the number of milliseconds until a transmission timeout will occur. If no transmission time
scheduled, this method returns zero.

clearTimeout, setRelativeTimeout:

(BOOL)resetAndEnable:(BOOL)enable

Does nothing and returns YES. Subclasses of IOTokenRing must implement this method so that it
the hardware. This method should invoke setRunning: to record the basic state of the device.

If enable is YES and the station is already in the ring, this method should do nothing but invoke se
argument and return YES. If enable is YES and the station isn't in the ring, interrupts should be en
inserted in the ring setRunning: should be used to update the device running status to YES or NO,
success of the insertion. If enable is NO, interrupts should be left disabled, the station should be re
and setRunning: should be invoked with a NO argument.

(unsigned int)ringSpeed

Returns the speed of the Token Ring, in megabits per second. This value, which is either 4 or 16, is specified by the ªRing Speedº key in the device configuration table. If the value is missing or inva to 16.

(void)setMaxInfoFieldSize:(unsigned int)size

Sets the maximum size of the info field. This value is used by allocateNetbuf. It's also used as the specified to the network subsystem. Your subclass should invoke this method in its implementatio initFromDeviceDescription:.

maxInfoFieldSize

(void)setRelativeTimeout:(unsigned int)timeout

Schedules a timeout to occur in timeout milliseconds. When timeout milliseconds pass without the (with clearTimeout), timeoutOccurred is invoked.

clearTimeout, relativeTimeout, timeoutOccurred (IODirectDevice)

(void)setRunning:(BOOL)running

Sets whether the hardware is inserted into the ring. The value of running should be YES to indicate inserted otherwise, it should be NO. This method is invoked only by methods in IOTokenRing sub IOTokenRing's own method implementations. You should invoke this method in your implementa resetAndEnable:.

isRunning

(BOOL)shouldAutoRecover

Returns YES if the device should try to recover from a failed attempt at inserting itself into the ring unexpected removal from the ring otherwise, returns NO. IOTokenRing sets this value depending ª Auto Recoveryº key in the device configuration table. This method is provided as a convenience subclasses that support automatic recovery.

(void)transmit:(netbuf_t)packet

Does nothing except free packet, using the nb_free() function. This method is invoked by the kerne subsystem when the hardware should transmit a packet.

Subclasses of IOTokenRing can implement this method or they can reimplement the method that i address:. To determine the number of bytes of data to be transmitted, use the nb_size() function. T data, use nb_map(). After getting the information you need from packet, you should free it with nb

outputPacket:address: (IONetworkDeviceMethods protocol)