NAME

ng_cisco - Cisco HDLC protocol netgraph node type

SYNOPSIS

#include <sys/types.h>
#include <netinet/in.h>
#include <netgraph/ng_cisco.h>

DESCRIPTION

The **cisco** node type performs encapsulation and de-encapsulation of packets using the Cisco HDLC protocol. This is a fairly simple protocol for the transmission of packets across high speed synchronous lines. Each packet is prepended with an Ethertype, indicating the protocol. There is also a "keep alive" and an "inquire" capability.

The downstream hook should connect to the synchronous line. On the other side of the node are the inet, inet6, atalk, and ipx hooks, which transmit and receive raw IP, IPv6, AppleTalk, and IPX packets, respectively. Typically these hooks would connect to the corresponding hooks on an ng_iface(4) type node.

IP Configuration

In order to function properly for IP traffic, the node must be informed of the local IP address and netmask setting. This is because the protocol includes an "inquire" packet which we must be prepared to answer. There are two ways to accomplish this, manually and automatically.

Whenever such an inquire packet is received, the node sends a NGM_CISCO_GET_IPADDR control message to the peer node connected to the inet hook (if any). If the peer responds, then that response is used. This is the automatic method.

If the peer does not respond, the node falls back on its cached value for the IP address and netmask. This cached value can be set at any time with a NGM_CISCO_SET_IPADDR message, and this is the manual method.

If the inet hook is connected to the inet hook of an ng_iface(4) node, as is usually the case, then configuration is automatic as the ng_iface(4) understands the NGM_CISCO_GET_IPADDR message.

HOOKS

This node type supports the following hooks:

downstream The connection to the synchronous line.

inet IP hook.

inet6 IPv6 hook.

atalk AppleTalk hook.

ipx IPX hook.

CONTROL MESSAGES

This node type supports the generic control messages, plus the following:

NGM_CISCO_SET_IPADDR (**setipaddr**)
This command takes an array of two struct in_addr arguments. The first is the IP address of the corresponding interface and the second is the netmask.

NGM_CISCO_GET_IPADDR (**getipaddr**)
This command returns the IP configuration in the same format used by NGM_CISCO_SET_IPADDR. This command is also *sent* by this node type to the inet peer whenever an IP address inquiry packet is received.

NGM_CISCO_GET_STATUS (**getstats**)
Returns a struct ngciscostat:

```
struct ngciscostat {
  uint32_t   seqRetries;        /* # unack'd retries */
  uint32_t   keepAlivePeriod;  /* in seconds */
};
```

SHUTDOWN

This node shuts down upon receipt of a NGM_SHUTDOWN control message, or when all hooks have been disconnected.

SEE ALSO

netgraph(4), ng_iface(4), ngctl(8)

D. Perkins, *Requirements for an Internet Standard Point-to-Point Protocol*, RFC 1547.

LEGAL

Cisco is a trademark of Cisco Systems, Inc.

HISTORY

The **ng_cisco** node type was implemented in FreeBSD 4.0.

AUTHORS

Julian Elischer <*julian@FreeBSD.org*>
Archie Cobbs <*archie@FreeBSD.org*>

BUGS

Not all of the functionality has been implemented. For example, the node does not support querying the remote end for its IP address and netmask.