

NAME

ng_async - asynchronous framing netgraph node type

SYNOPSIS

```
#include <sys/types.h>
#include <netgraph/ng_async.h>
```

DESCRIPTION

The **async** node type performs conversion between synchronous frames and asynchronous frames, as defined for the PPP protocol in RFC 1662. Asynchronous framing uses flag bytes and octet-stuffing to simulate a frame oriented connection over an octet-oriented asynchronous serial line.

The node transmits and receives asynchronous data on the **async** hook. Mbuf boundaries of incoming data are ignored. Once a complete packet has been received, it is decoded and stripped of all framing bytes, and transmitted out the **sync** hook as a single frame.

Synchronous frames are transmitted and received on the **sync** hook. Packets received on this hook are encoded as asynchronous frames and sent out on **async**. Received packets should start with the address and control fields, or the PPP protocol field if address and control field compression is employed, and contain no checksum field. If the first four bytes are 0xff 0x03 0xc0 0x21 (an LCP protocol frame) then complete control character escaping is enabled for that frame (in PPP, LCP packets are always sent with no address and control field compression and all control characters escaped).

This node supports "flag sharing" for packets transmitted on **async**. This is an optimization where the trailing flag byte of one frame is shared with the opening flag byte of the next. Flag sharing between frames is disabled after one second of transmit idle time.

HOOKS

This node type supports the following hooks:

async Asynchronous connection. Typically this hook would be connected to a **ng_tty(4)** node, which handles transmission of serial data over a tty device.

sync Synchronous connection. This hook sends and receives synchronous frames. For PPP, these frames should contain address, control, and protocol fields, but no checksum field. Typically this hook would be connected to an individual link hook of a **ng_ppp(4)** type node.

CONTROL MESSAGES

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

NGM_ASYNC_CMD_SET_CONFIG (setconfig)

Sets the node configuration, which is described by a struct `ng_async_cfg`:

```
struct ng_async_cfg {
    u_char  enabled; /* Turn encoding on/off */
    uint16_t amru; /* Max receive async frame length */
    uint16_t smru; /* Max receive sync frame length */
    uint32_t accm; /* ACCM encoding */
};
```

The `enabled` field enables or disables all encoding/decoding functions (default disabled). When disabled, the node operates in simple "pass through" mode. The `amru` and `smru` fields are the asynchronous and synchronous MRU (maximum receive unit) values, respectively. These both default to 1600; note that the async MRU applies to the incoming frame length after asynchronous decoding. The `accm` field is the asynchronous character control map, which controls the escaping of characters 0x00 through 0x1f (default 0xffffffff).

NGM_ASYNC_CMD_GET_CONFIG (getconfig)

This command returns the current configuration structure.

NGM_ASYNC_CMD_GET_STATS (getstats)

This command returns a struct `ng_async_stat` containing node statistics for packet, octet, and error counts.

NGM_ASYNC_CMD_CLR_STATS (clrstats)

Clears the node statistics.

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_ppp(4)`, `ng_tty(4)`, `ngctl(8)`

W. Simpson, *PPP in HDLC-link Framing*, RFC 1662.

W. Simpson, *The Point-to-Point Protocol (PPP)*, RFC 1661.

HISTORY

The `ng_async` node type was implemented in FreeBSD 4.0.

AUTHORS

Archie Cobbs <*archie@FreeBSD.org*>