NAME

ng_mppc - Microsoft MPPC/MPPE compression and encryption netgraph node type

SYNOPSIS

#include <sys/types.h>
#include <netgraph/ng_mppc.h>

DESCRIPTION

The **mppc** node type implements the Microsoft Point-to-Point Compression (MPPC) and Microsoft Point-to-Point Encryption (MPPE) sub-protocols of the PPP protocol. These protocols are often used in conjunction with the Point-to-Point Tunneling Protocol (PPTP).

The node has two hooks, comp for compression and decomp for decompression. Typically one or both of these hooks would be connected to the ng_ppp(4) node type hook of the same name. Each direction of traffic flow is independent of the other.

HOOKS

This node type supports the following hooks:

comp Connection to ng_ppp(4) comp hook. Incoming frames are compressed and/or encrypted, and sent back out the same hook.

decomp Connection to ng_ppp(4) decomp hook. Incoming frames are decompressed and/or decrypted, and sent back out the same hook.

CONTROL MESSAGES

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

NGM_MPPC_CONFIG_COMP

This command resets and configures the node for a session in the outgoing traffic direction (i.e., for compression and/or encryption). This command takes a struct ng_mppc_config as an argument:

```
/* Length of MPPE key */
#define MPPE_KEY_LEN      16

/* MPPC/MPPE PPP negotiation bits */
#define MPPC_BIT        0x00000001      /* mppc compression bits */
#define MPPE_40         0x00000020      /* use 40 bit key */
#define MPPE_56         0x00000080      /* use 56 bit key */
#define MPPE_128        0x00000040      /* use 128 bit key */
```

```
#define MPPE_BITS        0x000000e0     /* mppe encryption bits */
#define MPPE_STATELESS   0x01000000     /* use stateless mode */
#define MPPC_VALID_BITS  0x010000e1     /* possibly valid bits */

/* Configuration for a session */
struct ng_mppc_config {
  u_char    enable;              /* enable */
  uint32_t  bits;                /* config bits */
  u_char    startkey[MPPE_KEY_LEN]; /* start key */
};
```

The enabled field enables traffic flow through the node. The bits field contains the bits as negotiated by the Compression Control Protocol (CCP) in PPP. The startkey is only necessary if MPPE was negotiated, and must be equal to the session start key as defined for MPPE. This key is based on the MS-CHAP credentials used at link authentication time.

NGM_MPPC_CONFIG_DECOMP

This command resets and configures the node for a session in the incoming traffic direction (i.e., for decompression and/or decryption). This command takes a struct ng_mppc_config as an argument.

NGM_MPPC_RESETREQ

This message contains no arguments, and is bi-directional. If an error is detected during decompression, this message is sent by the node to the originator of the NGM_MPPC_CONFIG_DECOMP message that initiated the session. The receiver should respond by sending a PPP CCP Reset-Request to the peer.

This message may also be received by this node type when a CCP Reset-Request is received by the local PPP entity. The node will respond by flushing its outgoing compression and encryption state so the remote side can resynchronize.

SHUTDOWN

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

COMPILATION

The kernel options NETGRAPH_MPPC_COMPRESSION and NETGRAPH_MPPC_ENCRYPTION are supplied to selectively compile in either or both capabilities. At least one of these must be defined, or else this node type is useless.

SEE ALSO

netgraph(4), ng_ppp(4), ngctl(8)

G. Pall, *Microsoft Point-To-Point Compression (MPPC) Protocol*, RFC 2118.

G. S. Pall and G. Zorn, *Microsoft Point-To-Point Encryption (MPPE) Protocol*, draft-ietf-pppext-mppe-04.txt.

K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn, *Point-to-Point Tunneling Protocol (PPTP)*, RFC 2637.

AUTHORS

Archie Cobbs <*archie@FreeBSD.org*>

BUGS

In PPP, encryption should be handled by the Encryption Control Protocol (ECP) rather than CCP. However, Microsoft combined both compression and encryption into their ‘‘compression’’ algorithm, which is confusing.