NAME

ng_car - Committed Access Rate netgraph node type

SYNOPSIS

#include <netgraph/ng_car.h>

DESCRIPTION

The **car** node type limits traffic flowing through it using:

- Single rate three color marker as described in RFC 2697,
- Two rate three color marker as described in RFC 2698,
- RED-like rate limit algorithm used by Cisco,
- Traffic shaping with RED.

HOOKS

This node type supports the following hooks:

upper Hook leading to upper layer protocols.

lower Hook leading to lower layer protocols.

Traffic flowing from *upper* to *lower* is considered **downstream** traffic. Traffic flowing from *lower* to *upper* is considered **upstream** traffic.

MODES OF OPERATION

Each hook can operate in one of the following modes:

NG_CAR_SINGLE_RATE

Single rate three color marker as described in RFC 2697. Committed burst packets are counted as green, extended burst packets are counted as yellow and exceeding packets are counted as red. Committed burst getting refilled with CIR (Committed Information Rate) speed. When it is full, exceeded burst getting refilled.

NG_CAR_DOUBLE_RATE

Two rate three color marker as described in RFC 2698. Committed burst packets are counted as green, peak burst packets are counted as yellow and exceeding packets are counted as red. Committed burst getting refilled with CIR speed. Peak burst getting refilled with PIR (Peak Information Rate) speed at the same time.

NG_CAR_RED

Similar to NG_CAR_SINGLE_RATE, but with different understanding of extended burst. When normal burst exceeded and extended burst is used, packets are counted red with probability equal to part of extended burst consumed. Extended burst getting refilled first. When it is full, committed burst getting refilled. This behavior is similar to RED active queue management algorithm.

This algorithm is more polite to the TCP traffic than NG_CAR_SINGLE_RATE.

NG_CAR_SHAPE
: Committed burst packets are counted as green, exceeding packets are delayed by queue with RED management and counted as yellow. Packets dropped by queue counted as red. Queue parameters are hardcoded: length 99 packets, min_th 8 packets, max_p 100%.

Traffic shaping is much more polite to the TCP traffic than rate limit on links with bandwidth * delay product less than 6-8 TCP segments, but it consumes additional system resources for queue processing.

By default, all information rates are measured in bits per second and bursts are measured in bytes. But when NG_CAR_COUNT_PACKETS option is enabled, rates are measured in packets per second and bursts are in packets.

CONTROL MESSAGES

This node type supports the generic control messages and the following specific messages.

NGM_CAR_SET_CONF (**setconf**)
: Set node configuration to the specified at *struct ng_car_bulkconf*

NGM_CAR_GET_CONF (**getconf**)
: Return current node configuration as *struct ng_car_bulkconf*

```
struct ng_car_hookconf {
        uint64_t cbs;             /* Committed burst size (bytes) */
        uint64_t ebs;             /* Exceeded/Peak burst size (bytes) */
        uint64_t cir;             /* Committed information rate (bits/s) */
        uint64_t pir;             /* Peak information rate (bits/s) */
        uint8_t  green_action;    /* Action for green packets */
        uint8_t  yellow_action;   /* Action for yellow packets */
        uint8_t  red_action; /* Action for red packets */
        uint8_t  mode;            /* single/double rate, ... */
        uint8_t  opt;             /* color-aware or color-blind */
};
```

```
/* possible actions (..._action) */
enum {
    NG_CAR_ACTION_FORWARD = 1,
    NG_CAR_ACTION_DROP,
    NG_CAR_ACTION_MARK
};

/* operation modes (mode) */
enum {
    NG_CAR_SINGLE_RATE = 0,
    NG_CAR_DOUBLE_RATE,
    NG_CAR_RED,
    NG_CAR_SHAPE
};

/* mode options (bits for opt) */
#define NG_CAR_COLOR_AWARE   1
#define NG_CAR_COUNT_PACKETS         2

struct ng_car_bulkconf {
        struct ng_car_hookconf upstream;
        struct ng_car_hookconf downstream;
};
```

NGM_CAR_GET_STATS (**getstats**)

Return node statistics as *struct ng_car_bulkstats*

```
struct ng_car_hookstats {
        uint64_t passed_pkts;       /* Counter for passed packets */
        uint64_t dropped_pkts;      /* Counter for dropped packets */
        uint64_t green_pkts;        /* Counter for green packets */
        uint64_t yellow_pkts;       /* Counter for yellow packets */
        uint64_t red_pkts;   /* Counter for red packets */
        uint64_t errors;     /* Counter for operation errors */
};

struct ng_car_bulkstats {
        struct ng_car_hookstats upstream;
        struct ng_car_hookstats downstream;
};
```

NGM_CAR_CLR_STATS (**clrstats**)
Clear node statistics.

NGM_CAR_GETCLR_STATS (**getclrstats**)
Atomically return and clear node statistics.

SHUTDOWN

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

EXAMPLES

Limit outgoing data rate over fxp0 Ethernet interface to 20Mbit/s and incoming packet rate to 5000pps.

```
/usr/sbin/ngctl -f- <<-SEQ
        mkpeer fxp0: car lower lower
        name fxp0:lower fxp0_car
        connect fxp0: fxp0_car: upper upper
        msg fxp0_car: setconf { downstream={ cir=20000000 cbs=2500000 ebs=2500000 greenAction=1 yellowA
SEQ
```

SEE ALSO

netgraph(4), ngctl(8)

J. Heinanen, *A Single Rate Three Color Marker*, RFC 2697.

J. Heinanen, *A Two Rate Three Color Marker*, RFC 2698.

AUTHORS

Nuno Antunes <*nuno.antunes@gmail.com*>
Alexander Motin <*mav@FreeBSD.org*>

BUGS

At this moment only DROP and FORWARD actions are implemented.