

NAME

h_ertt - Enhanced Round Trip Time Khelp module

SYNOPSIS

```
#include <netinet/khelp/h_ertt.h>
```

DESCRIPTION

The **h_ertt** Khelp module works within the khelp(9) framework to provide TCP with a per-connection, low noise estimate of the instantaneous RTT. The implementation attempts to be robust in the face of delayed acknowledgements, TCP Segmentation Offload (TSO), receivers who manipulate TCP timestamps and lack of the TCP timestamp option altogether.

TCP receivers using delayed acknowledgements either acknowledge every second packet (reflecting the time stamp of the first) or use a timeout to trigger the acknowledgement if no second packet arrives. If the heuristic used by **h_ertt** determines that the receiver is using delayed acknowledgements, it measures the RTT using the second packet (the one that triggers the acknowledgement). It does not measure the RTT if the acknowledgement is for the first packet, since it cannot be accurately determined.

When TSO is in use, **h_ertt** will momentarily disable TSO whilst marking a packet to use for a new measurement. The process has negligible impact on the connection.

h_ertt associates the following struct with each connection's TCP control block:

```
struct ertt {
    TAILQ_HEAD(txseginfo_head, txseginfo) txsegi_q;    /* Private. */
    long          bytes_tx_in_rtt;                    /* Private. */
    long          bytes_tx_in_marked_rtt;
    unsigned long marked_snd_cwnd;
    int           rtt;
    int           maxrtt;
    int           minrtt;
    int           dlyack_rx;                          /* Private. */
    int           timestamp_errors;                   /* Private. */
    int           markedpkt_rtt;                      /* Private. */
    uint32_t     flags;
};
```

The fields marked as private should not be manipulated by any code outside of the **h_ertt** implementation. The non-private fields provide the following data:

<i>bytes_tx_in_marked_rtt</i>	The number of bytes transmitted in the <i>markedpkt_rtt</i> .
<i>marked_snd_cwnd</i>	The value of cwnd for the marked rtt measurement.
<i>rtt</i>	The most recent RTT measurement.
<i>maxrtt</i>	The longest RTT measurement that has been taken.
<i>minrtt</i>	The shortest RTT measurement that has been taken.
<i>flags</i>	The ERTT_NEW_MEASUREMENT flag will be set by the implementation when a new measurement is available. It is the responsibility of h_ertt consumers to unset the flag if they wish to use it as a notification method for new measurements.

SEE ALSO

`cc_chd(4)`, `cc_hd(4)`, `cc_vegas(4)`, `mod_cc(4)`, `hhook(9)`, `khel(9)`

ACKNOWLEDGEMENTS

Development and testing of this software were made possible in part by grants from the FreeBSD Foundation and Cisco University Research Program Fund at Community Foundation Silicon Valley.

HISTORY

The **h_ertt** module first appeared in FreeBSD 9.0.

The module was first released in 2010 by David Hayes whilst working on the NewTCP research project at Swinburne University of Technology's Centre for Advanced Internet Architectures, Melbourne, Australia. More details are available at:

<http://caia.swin.edu.au/urp/newtcp/>

AUTHORS

The **h_ertt** Khel module and this manual page were written by David Hayes <david.hayes@ieee.org>.

BUGS

The module maintains enhanced RTT estimates for all new TCP connections created after the time at which the module was loaded. It might be beneficial to see if it is possible to have the module only affect connections which actually care about ERTT estimates.