Sender RTT Estimate Option
for the Datagram Congestion Control Protocol (DCCP)

Abstract

   This document specifies an update to the round-trip time (RTT)
   estimation algorithm used for TFRC (TCP-Friendly Rate Control)
   congestion control by the Datagram Congestion Control Protocol
   (DCCP). It updates specifications for the CCID-3 and CCID-4
   Congestion Control IDs of DCCP.

   The update addresses parameter-estimation problems occurring with
   TFRC-based DCCP congestion control. It uses a recommendation made in
   the original TFRC specification to avoid the inherent problems of
   receiver-based RTT sampling, by utilising higher-accuracy RTT samples
   already available at the sender.

   It is integrated into the feature set of DCCP as an end-to-end
   negotiable extension.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6323.

Copyright Notice

Table of Contents

1. Introduction

   The Datagram Congestion Control Protocol (DCCP) [RFC4340] is a
   transport protocol for connection-oriented, unreliable, and
   congestion-controlled datagram delivery. In DCCP, an application has
   a choice of congestion control mechanisms, each specified by a
   Congestion Control Identifier (CCID; [RFC4340], Section 10).

   This document defines a Standards-Track update to the sender and
   receiver sides of two rate-based DCCP congestion control IDs: CCID-3
   [RFC4342] and the Experimental CCID-4 variant [RFC5622].

   Both CCIDs are based on the principles of TCP-Friendly Rate Control
   (TFRC) [RFC5348], which performs rate-based congestion control. Its
   feedback mechanism differs from that used by window-based congestion
   control such as in TCP. As a consequence, in TFRC the feedback may
   be sent less frequently (e.g., once per round-trip time).
   Furthermore, a measured RTT estimate is directly used as the basis
   for computing the (TCP-friendly) transmission rate.

   In TFRC-based protocols, packets are rate-paced over an RTT, instead
   of allowing them to be sent back-to-back as they could be in TCP;
   thus, accurate RTT estimation is important to ensure appropriate
   pacing at the sender.

   The original specifications for CCID-3 and CCID-4, in [RFC4342] and
   [RFC5622], both estimate the RTT at the receiver, using an algorithm
   based on the cyclic 4-bit window counter of the DCCP CCVal header.
   The method has implications that have been observed when using
   applications over DCCP implementations, resulting in infrequent and
   inaccurate RTT measurement.

   This update addresses these RTT estimation problems by providing a
   solution based on a concept first recommended in [RFC5348], Section
   3.2.1; i.e., to measure the RTT at the sender. That approach results
   in a higher reliability and frequency of samples and avoids the
   inherent problems of receiver-based RTT sampling discussed below.

   The document begins by analysing the encountered problems in the next
   section. The update is presented in Section 3. We then discuss
   security considerations in Section 4 and list the resulting IANA
   considerations in Section 5.

2. Problems Caused by Sampling the RTT at the Receiver

   There are at least six areas that make a TFRC receiver vulnerable to
   inaccuracies or absence of (receiver-based) RTT samples:

   o the measured sending rate, X_recv ([RFC5348], Section 6.2);

   o synthesis of the first loss interval ([RFC5348], Section 6.3.1);

   o disambiguation of loss events ([RFC4342], Section 10.2);

   o validation of loss intervals ([RFC4342], Section 6.1);

   o ensuring that at least one feedback packet is sent per RTT
      ([RFC4342], Section 10.3);

   o determining quiescence periods ([RFC4342], Section 6.4).

2.1. List of Problems Encountered with a Real Implementation

   This section summarizes several years of experience using the Linux
   implementation of CCID-3 and CCID-4. It lists the problems
   encountered with receiver-based RTT sampling over real networks, in a
   variety of wired and wireless environments and under different link-
   layer conditions.

   The Linux DCCP/TFRC implementation is based on the RTT-sampling
   algorithm specified in [RFC4342], Section 8.1. This algorithm relies
   on a coarse-grained window counter (units of RTT/4), and uses packet
   inter-arrival times to estimate the current RTT of the network.

   The algorithm is effective only for packets with modulo-16 CCVal
   differences less than 5, due to limitations noted in Sections 8.1 and
   10.3 of [RFC4342]. A CCVal difference less than 4 means sampling at
   sub-RTT scale; [RFC4342], Section 8.1 thus suggests differences
   between 2 and 4, the latter being preferable (equivalent to a full
   RTT). The same section limits the maximum CCVal difference between
   data-carrying packets to 5, in order to avoid wrap-around. As a
   consequence, it is not possible to determine the timing interval for
   adjacent packets with a CCVal difference greater than 4: such samples
   have to be discarded.

   A second problem arises when there are holes in the sequence space.
   Because the 4-bit CCVal counter may cycle around multiple times, it
   is not possible to determine window-counter wrap-around whenever
   sequence numbers of subsequent packets are not immediately adjacent.
   This problem occurs when packets are delayed, reordered, or lost in
   the network.

As a result, RTT sampling has to be paused during times of loss.
However, this aggravates the problem, since the sender now requires
new feedback from the receiver, but the receiver is unable to provide
accurate and up-to-date information: the receiver is unable to sample
the RTT, and accordingly is also unable to estimate X_recv correctly,
which then in turn affects X_Bps at the sender.

The third limitation arises from using inter-arrival times as
representatives of network inter-packet gaps. It is well known that
the inter-packet gap of packets is not constant along a network path.
Furthermore, modern network interface cards do not necessarily
deliver each packet at the time it is received, but rather in a
bunch, to avoid overly frequent interrupts [MR97]. As a result,
inter-packet arrival times may converge to zero, when subsequent
packets are being delivered at virtually the same time.

The fourth problem is that of under-sampling and thus related to the
first limitation. If loss occurs while the receiver has not yet had
a chance to sample the RTT, it needs to fall back to some fixed RTT
constant to plug into the equation of [RFC5348], Section 6.3.1. (The
sender, for example, uses a fixed value of 1 second when it is unable
to obtain an initial RTT sample; see [RFC5348], Section 4.2).

In particular, if the loss is caused by a transient condition, this
fourth problem causes a subsequent deterioration of the connection
(rate reduction), further aggravated by the fact that TFRC takes
longer than common window-based protocols to recover from a reduction
of its allowed sending rate.

Trying to smooth over these effects by imposing heavy filtering on
the RTT samples did not substantially improve the situation, nor does
it solve the problem of under-sampling.

The TFRC sender, on the other hand, is much better equipped to
estimate the RTT and can do this more accurately. This is in
particular due to the use of timestamps and elapsed time information
([RFC5348], Section 3.2.2), which are mandatory in CCID-3 (Sections 6
and 8.2 of [RFC4342]).

2.2. Other Areas Affected by the RTT Sampling Problems

   Here we analyse the impact that unreliability of receiver-based RTT
   sampling has on the areas listed at the beginning of Section 2.

   In addition, benefits of sender-based RTT sampling have already been
   pointed out in [RFC5348] and in the specification of CCID-3 at the
   end of Section 10.2 of [RFC4342].

2.2.1. Measured Receive Rate X_recv

   A key problem is that the reliability of X_recv [RFC4342] depends
   directly upon the reliability and accuracy of RTT samples. This
   means that failures propagate from one parameter to another.

   Errata IDs 610 [Err610] and 611 [Err611] update [RFC4342] to use the
   definition of the receive rate as specified in [RFC5348].

   Having an explicit (rather than a coarse-grained) RTT estimate allows
   measurement of X_recv with greater accuracy and isolates failure.

   An explicit RTT estimate also enables the receiver to more accurately
   perform the test in step (2) of [RFC4342], Section 6.2, i.e., to
   check whether less or more than one RTT has passed since the last
   feedback.

2.2.2. Disambiguation and Accuracy of Loss Intervals

   Since a loss event is defined as one or more data packets in one RTT
   that are lost or marked with Explicit Congestion Notification (ECN;
   [RFC5348], Section 5.2), the receiver needs accurate RTT estimates to
   validate and accurately separate loss events. Moreover, Section 5.2
   of [RFC5348] expressly indicates the sender RTT estimate is
   RECOMMENDED for this purpose.

   Having the sender RTT Estimate available further increases the
   accuracy of the information reported by the receiver. The definition
   of Loss Intervals in [RFC4342], Section 6.1 needs the RTT to separate
   the lossy parts; in particular, lossy parts spanning a period of more
   than one RTT are invalid.

   A similar benefit arises in the computation of the loss event rate:
   as discussed in Section 9.2 of [RFC4342], it may happen that the
   sender and receiver compute different loss event rates, due to
   differences in the available timing information. An explicit RTT
   estimate increases the accuracy of information available at the
   receiver; thus, the sender may not need to recompute the (less
   reliable) loss event rate reported by the receiver.

2.2.3. Determining Quiescence

   The quiescence period is defined as max(2 * RTT, 0.2 sec) in Section
   6.4 of [RFC4342]. An explicit RTT estimate avoids under- and over-
   estimating quiescence periods.

2.2.4. Practical Considerations

   Using explicit RTT estimates contributes to greater robustness and
   can also result in simpler implementation.

   First, it becomes easier to separate adjacent loss events. The 4-bit
   counter value wraps relatively frequently, which requires additional
   procedures to avoid aliasing effects.

   Second, the receiver is better able to determine when to send
   feedback packets. It can perform the test described in step (2) of
   [RFC5348], Section 6.2 more accurately. Moreover, unnecessary
   expiration of the nofeedback timer (as described in [RFC4342],
   Section 10.3) can be avoided.

   Lastly, a sender-based RTT estimate option can be used by middleboxes
   to verify that a flow uses conforming end-to-end congestion control
   ([RFC4342], Section 10.2).

3. Specification

3.1. Conventions

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119].

   This document uses the conventions of [RFC5348], [RFC4340],
   [RFC4342], and [RFC5622].

   All multi-byte field descriptions presented in this document are in
   network byte order (most significant byte first).

3.2. Options and Features

   This document defines a single TFRC-specific option, RTT Estimate,
   described in the next subsection.

   Following the guidelines in [RFC4340], Section 15, the use of the RTT
   Estimate Option is governed by an associated feature, Send RTT
   Estimate Feature. This feature is described in Section 3.2.2.
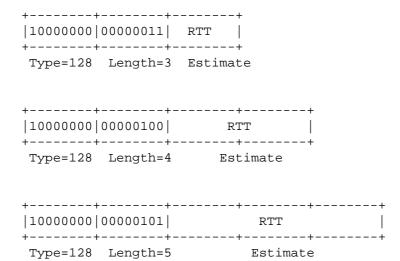
3.2.1. RTT Estimate Option

   The sender communicates its current RTT estimate to the receiver
   using an RTT Estimate Option.

```
     +------+--------------+-------------+------------+
     | Type | Option Length |   Meaning  | DCCP Data? |
     +------+--------------+-------------+------------+
     | 128  |     3/4/5    | RTT Estimate |     Y     |
     +------+--------------+-------------+------------+
```

             Table 1: The RTT Estimate Option Defined by This Document

   Column meanings are as per [RFC4340], Section 5.8 (Table 3). This
   option MAY be placed in any DCCP packet, has option number 128 and a
   length of 3..5 bytes.

   A Sender RTT Estimate Option is valid if it satisfies one of the
   three following formats:

```
     +--------+--------+--------+
     |10000000|00000011|  RTT   |
     +--------+--------+--------+
      Type=128  Length=3  Estimate
```

```
     +--------+--------+--------+--------+
     |10000000|00000100|      RTT        |
     +--------+--------+--------+--------+
      Type=128  Length=4      Estimate
```

```
     +--------+--------+--------+--------+--------+
     |10000000|00000101|          RTT            |
     +--------+--------+--------+--------+--------+
      Type=128  Length=5          Estimate
```

   The 1..3 value bytes of the option data carry the current RTT
   estimate of the sender, using a granularity of 1 microsecond. This
   allows values up to 16.7 seconds (corresponding to 0xFFFFFE) to be
   communicated.

   A sender capable of sampling at sub-microsecond granularity SHOULD
   round up RTT samples to the next microsecond, to avoid under-
   estimating the RTT.

   The value 0xFFFFFF is reserved to indicate significant delay spikes,
   larger than 16.7 seconds. This is qualitative rather than
   quantitative information, to alert the receiver that there is a
   network problem (for instance, jamming on a wireless channel).

The use of the RTT Estimate Option on networks with RTTs larger than
16.7 seconds is not specified by this document (as per Section 3.3,
the sender would then always report 0xFFFFFF).

A value of 0 indicates the absence of a valid RTT sample. The sender
MUST set the value to 0 if it does not yet have an RTT estimate. RTT
estimates of less than 1 microsecond MUST be reported as 1
microsecond.

The sender SHOULD select the smallest format suitable to carry the
RTT estimate (i.e., less than 1 byte of leading zeroes).

3.2.2. Send RTT Estimate Feature

The Send RTT Estimate feature lets endpoints negotiate whether the
sender MUST provide RTT Estimate options on its data packets.

Send RTT Estimate has feature number 128 and is server-priority. It
takes 1-byte Boolean values; values greater than 1 are reserved.

```
+--------+-------------------+------------+---------------+-------+
| Number |      Meaning      | Rec'n Rule | Initial Value | Req'd |
+--------+-------------------+------------+---------------+-------+
|   128  | Send RTT Estimate |     SP     |       0       |   N   |
+--------+-------------------+------------+---------------+-------+
```

Table 2: The Send RTT Estimate Feature Defined by This Document

The column meanings are described in [RFC4340], Section 6.4.

The Send RTT Estimate feature is OPTIONAL. An extension may
implement it, but this specification does not require the feature to
be understood by every DCCP implementation (see [RFC4340], Section
15). The feature is off by default (initial value of 0).

DCCP B sends a "Mandatory Change R(Send RTT Estimate, 1)" to require
DCCP A to send RTT Estimate options as part of its data traffic (DCCP
A will reset the connection if it does not understand this feature).

3.3. Basic Usage

When the Send RTT Estimate Feature is enabled, the sender MUST
provide an RTT Estimate Option on all of its Data, DataAck, Sync, and
SyncAck packets. It MAY in addition provide the RTT Estimate Option
on other packet types, such as DCCP-Ack. If the RTT is larger than
the maximum representable value (0xFFFFFE), the sender MUST set the
value of the RTT Estimate Option to 0xFFFFFF.

The sender MUST implement and continue to update the CCVal window
counter as specified in [RFC4342], Section 8.1, even when the Send
RTT Estimate Feature is on.

When the Send RTT Estimate Feature is enabled, the receiver MUST use
the value reported by the RTT Estimate Option in all places that
require an RTT (listed at the begin of Section 2). If the receiver
encounters an invalid RTT Estimate Option (Section 3.2.1), it MUST
reset the connection with Reset Code 5, "Option Error", where the
Data 1..3 fields are set to the first 3 bytes of the offending RTT
Estimate Option.

The receiver SHOULD track the long-term RTT estimate using a moving
average, such as the one specified in [RFC5348], Section 4.3. This
long-term estimate is referred to as "receiver_RTT" below.

When the Send RTT Estimate Feature is disabled, the receiver MUST
estimate the RTT as previously specified in [RFC4340], [RFC4342], and
[RFC5622].

3.4. Receiver Robustness Measures

This subsection specifies robustness measures for the receiver when
the Send RTT Estimate Feature is on.

The 0-valued and 0xFFFFFF-valued RTT Estimate Options are both
referred to as "no-number RTT options". RTT Estimate Options with
values in the range of 1..0xFFFFFE are analogously called "numeric
RTT options".

Until the first numeric RTT option arrives, the receiver MUST use a
value of 0.5 seconds for receiver_RTT (to match the initial 2-second
timeout of the TFRC nofeedback timer; see [RFC5348], Section 4.2).

If the path RTT is known, e.g., from a previous connection [RFC2140],
the receiver MAY reuse the previously known path RTT value to seed
its long-term RTT estimate.

The sender MAY occasionally send no-number RTT options, covering for
transient changes and spurious disruptions. During these times, the
receiver SHOULD continue to use its long-term receiver_RTT value.

To avoid under-estimating the RTT in the absence of numeric options,
the receiver MUST back off receiver_RTT in the following manner: if
the sender supplies no-number RTT options for longer than
receiver_RTT units of time, the receiver sets

        receiver_RTT = MIN(2 * receiver_RTT, t_mbi)

   where t_mbi = 64 seconds is the maximum back-off interval ([RFC5348],
   Appendix A). For the next round of no-number RTT options, the
   updated value of receiver_RTT applies.

   This back-off mechanism ensures that short-term disruptions do not
   have a lasting impact, whereas long-term problems will result in
   asymptotically high receiver_RTT values.

   To bail out from a hanging session, the receiver MAY close the
   connection when receiver_RTT has reached the value MAX_RTT.

4. Security Considerations

   Security considerations for CCID-3 have been discussed in Section 11
   of [RFC4342]; for CCID-4, these have been discussed in Section 13 of
   [RFC5622], referring back to the same section of [RFC4342].

   This document introduces an extension to communicate the current RTT
   estimate of the sender to the receiver of a TFRC communication.

   By altering the value of the RTT Estimate Option, it is possible to
   interfere with the behaviour of a flow using TFRC. In particular,
   since accuracy of the RTT estimate directly influences the accuracy
   of the measured sending rate X_recv, it would be possible to obtain
   either higher or lower sending rates than are warranted by the
   current network conditions.

   This is only possible if an attacker is on the same path as the DCCP
   sender and receiver, and is able to guess valid sequence numbers.
   Therefore, the considerations in Section 18 of [RFC4340] apply.

5. IANA Considerations

   This document requests identical allocation in the dccp-ccid3-
   parameters and the dccp-ccid4-parameters registries.

5.1. Option Types

   This document defines a single CCID-specific option (128) for
   communicating RTT estimates from the HC-sender to the HC-receiver.
   Following [RFC4340], Section 10.3, this requires an option number for
   the RTT Estimate Option in the range 128..191.

5.2. Feature Numbers

   This document defines a single CCID-specific feature number (128) for
   the Send RTT Estimate feature, which is located at the HC-sender.
   Following [RFC4340], Section 10.3, a feature number in the range
   128..191 is required.

6. References

6.1. Normative References

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC4340]   Kohler, E., Handley, M., and S. Floyd, "Datagram
               Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

   [RFC4342]   Floyd, S., Kohler, E., and J. Padhye, "Profile for
               Datagram Congestion Control Protocol (DCCP) Congestion
               Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342,
               March 2006.

   [RFC5348]   Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP
               Friendly Rate Control (TFRC): Protocol Specification",
               RFC 5348, September 2008.

   [RFC5622]   Floyd, S. and E. Kohler, "Profile for Datagram Congestion
               Control Protocol (DCCP) Congestion ID 4: TCP-Friendly Rate
               Control for Small Packets (TFRC-SP)", RFC 5622,
               August 2009.

6.2. Informative References

   [Err610]   RFC Errata, Errata ID 610, RFC 4342,
               <http://www.rfc-editor.org>.

   [Err611]   RFC Errata, Errata ID 611, RFC 4342,
               <http://www.rfc-editor.org>.

   [MR97]   Mogul, J. and K. Ramakrishnan, "Eliminating Receive
               Livelock in an Interrupt-Driven Kernel", ACM Transactions
               on Computer Systems (TOCS), 15(3):217-252, August 1997.

   [RFC2140]   Touch, J., "TCP Control Block Interdependence", RFC 2140,
               April 1997.

Authors' Addresses

    Gerrit Renker
    University of Aberdeen
    School of Engineering
    Fraser Noble Building
    Aberdeen AB24 3UE
    Scotland

    EMail: gerrit@erg.abdn.ac.uk
    URI: http://www.erg.abdn.ac.uk

    Godred Fairhurst
    University of Aberdeen
    School of Engineering
    Fraser Noble Building
    Aberdeen AB24 3UE
    Scotland

    EMail: gorry@erg.abdn.ac.uk
    URI: http://www.erg.abdn.ac.uk