Internet Engineering Task Force
INTERNET-DRAFT
draft-padhye-dcp-ccid3-00.ps

Jitendra Padhye
Sally Floyd
Eddie Kohler
ACIRI

13 July 2001
Expires: January 2002

Profile for DCP Congestion Control ID 3: TFRC Congestion Control

Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document contains the profile for Congestion Control Identifier 3, TCP-friendly rate control (TFRC), in the Datagram Control Protocol (DCP). DCP implements a congestion-controlled unreliable datagram flow suitable for use by applications such as streaming media. The TFRC CCID is used by applications that want a TCP-friendly send rate, possibly with Explicit Congestion Notification (ECN), while minimizing abrupt rate changes.

Table of Contents

1. Introduction 3
1.1. Usage Scenario. 3
1.2. Example Half-Connection 3
2. Connection Establishment 4
3. Congestion Control on Data Packets 4
4. Acknowledgments 4
4.1. Congestion Control on Acknowledgments 5
4.2. Quiescence 5
4.3. Acknowledgments of Acknowledgments 5
5. Explicit Congestion Notification 5
6. Relevant Options and Features 5
6.1. RTT Option. 5
6.2. Send Rate Option 5
6.3. Loss Event Rate Option 6
6.4. Receive Rate Option 6
6.5. ECN NONCE Option 6
7. Application Requirements 7
8. Thanks 7
9. References 7
10. Authors' Addresses 8

1. Introduction

This document contains the profile for Congestion Control Identifier 3, TCP-friendly rate control (TFRC), in the Datagram Control Protocol (DCP). DCP uses Congestion Control Identifiers, or CCIDs, to specify the congestion control mechanism in use on a half-connection. (A half-connection might consist of data packets sent from DCP A to DCP B, plus acknowledgments sent from DCP B to DCP A. DCP A is the sending DCP, and DCP B the acknowledging DCP, for this half-connection.)

TFRC is a receiver-based congestion control mechanism that provides a TCP-friendly send rate, while minimizing abrupt rate changes [1].

The basic TFRC protocol is as follows. The sender sends a stream of data packets to the receiver at some rate. The receiver sends a feedback packet to the sender once every round-trip time. Based on the information contained in the feedback packets, the sender adjusts its sending rate in accordance with the TCP throughput equation [2], to maintain TCP-friendliness. If no feedback is received from the receiver in two round-trip times, the sender halves its sending rate.

The values of the round-trip time (*RTT*), the loss event rate (p) and the base timeout value *T_0* are needed to calculate the send rate using the TCP throughput equation. In the TFRC protocol, the sender is responsible for calculating the values of *RTT* and *T_0*, while the receiver is responsible for calculating the value of p.

We note that this draft is a first pass, and is not necessarily complete.

1.1. Usage Scenario

TFRC is intended to provide congestion control for the flow of data packets from the server to the client for applications that do not require fully reliable data transmission.

1.2. Example Half-Connection

This example, taken from the main DCP draft, is of a half-connection using TFRC Congestion Control specified by CCID 3. The "sender" is the HC-Sender, and the "receiver" is the HC-Receiver.

1 The server sends DCP-Data packets, where the number of packets sent is governed by an allowed transmit rate, as specified in [1]. Each DCP-Data packet has a sequence number, and includes an Acknowledgment Number that is the sequence number of the most recent acknowledgment packet received from the client. Each DCP-Data packet also contains a timestamp, the server's estimate of the round-trip time, and the current sending rate.

One or more of these data packets are DCP-DataAck packets acknowledging the data packet from the client, but for simplicity we will not discuss the half-connection of data from the client to the server in this example.

2 The client sends DCP-Ack packets at least once per round-trip time acknowledging the data packets, or as indicated by the TFRC specification [1]. Each DCP-Ack packet uses a sequence number and identifies the most recent packet received from the server. Each DCP-Ack packet includes feedback about the loss event rate calculated by the client, as specified below.

3 The server continues sending DCP-Data packets as controlled by the allowed transmit rate. Upon receiving DCP-Ack packets, the server updates its allowed transmit rate as specified in [1].

4 The server estimates round-trip times and calculates a TimeOut (TO) value as specified in [1].

5 Each DCP-Data, DCP-DataAck, and DCP-Ack packet is sent as ECN-Capable, with either the ECT(0) or the ECT(1) codepoint set. The use of the ECN Nonce with TFRC is described below.

2. Connection Establishment

The connection is initiated by the client using mechanisms described in the DCP specification [3]. The client and the server MAY negotiate the use of ACK Vector option. Both the server and the client MUST support the timestamp option. The ACK vector option and the timestamp option are described in [3].

3. Congestion Control on Data Packets

The sender sends DCP-Data packets to the receiver at the rate specified by the TCP throughput equation [2].

Each DCP-Data packet has a sequence number, and an acknowledgment number that is the sequence number of the most recent acknowledgment packet received from the receiver. Each data packet contains the following options:

1. Timestamp option, as described in [3].
2. An option specifying sender's estimate of the round-trip time.
3. An option specifying the current sending rate.

The format of options 2 and 3 is described below.

After each feedback packet is received from the receiver, the sender updates values of *RTT*, *T_0* and the sending rate using procedures specified in [1].

If no feedback packet is received from the receiver, the sending rate is halved. However, the sending rate is never reduced below one packet per 64 seconds. See [1] for more details.

4. Acknowledgments

The receiver sends DCP-Ack packets to the sender once per round-trip time, or more frequently. This rate is determined by details of the TFRC protocol, as specified in [1].

The acknowledgment number in DCP-Ack acknowledges the most recent packet received from the sender. Each Ack packet from the receiver includes the following options:

1. The echo timestamp option described in [3].
2. An option specifying the loss event rate (p) calculated by the receiver as described in [1].
3. An option specifying the rate at which the receiver received data since the last DCP-Ack was sent.

The format of options 2 and 3 is described below.

4.1. Congestion Control on Acknowledgments

The rate and timing for generating acknowledgments is determined by the TFRC algorithm [1].

4.2. Quiescence

This section refers to quiescence in the DCP sense (see section 6.1 of [3]): How does a CCID 3 receiver determine that the corresponding sender is not sending any data?

The receiver detects that the sender has gone quiescent after two round-trip times have passed without receiving any additional data. Since ACKs are not required to be reliable, the receiver needs to do nothing special in this case, unlike CCID 2 [5].

4.3. Acknowledgments of Acknowledgments

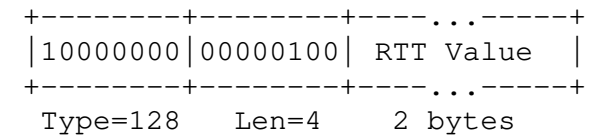
Acknowledgments in TFRC are entirely unreliable -- TFRC works even if every acknowledgment is dropped -- and it is never necessary for the sender to acknowledge an acknowledgment.

5. Explicit Congestion Notification

ECN [6] MAY be used with CCID 3. If ECN is used, then the ECN Nonce will automatically be used for the data packets, following the specification for the ECN Nonce [4] for TCP. For the data sub-flow, the sender sets either the ECT[0] or ECT[1] codepoint on DCP-Data packets. If the ACK vector option is being used, the ECN-NONCE information is returned via the ACK vector. Otherwise, the information about ECN-NONCE is returned by the receiver using the ECN-NONCE option described below. The receiver MUST return this option if ECN is being used, and if it is reporting a lower packet loss rate than the one it reported in the previous acknowledgment. We are working on this further.

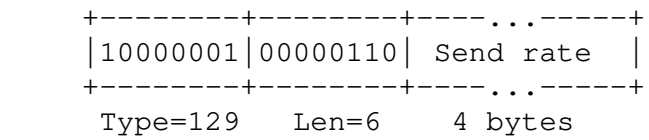
6. Relevant Options and Features

6.1. RTT Option



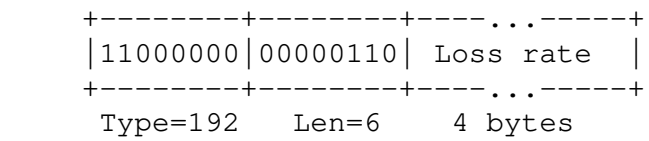
This option is set by the data sender on all data packets. The first byte gives the option type and the second gives the option length. The last two bytes indicate sender's estimation of round-trip time in whole milliseconds.

6.2. Send Rate Option



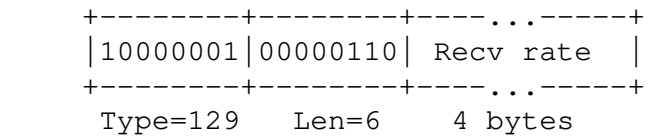
This option is set by the data sender on all data packets. The first byte gives the option type and the second gives the option length. The last four bytes indicate the current sending rate in bits per second.

6.3. Loss Event Rate Option



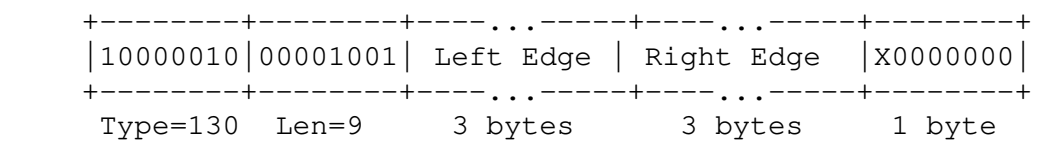
This option is set by the data receiver on all acknowledgment packets. The first byte gives the option type and the second gives the option length. The last four bytes indicate the inverse of the loss event rate, rounded UP, as calculated by the receiver.

6.4. Receive Rate Option



This option is set by the data receiver on all acknowledgment packets. The first byte gives the option type and the second gives the option length. The last four bytes indicate the the rate at which the receiver received the data since it last sent the acknowledgment, in bits per second.

6.5. ECN NONCE Option



This option is set by the data receiver on any acknowledgment packet that reports a loss rate lower than the loss rate reported in the previous acknowledgment packet. The first byte gives the option type and the second gives the option length. The right edge (RE) and the left edge (LE) are sequence numbers of data packets, such that:

- Let LastAck be the sequence number of the data packet acknowledged by the previous acknowledgment.

- If (LastAck + 1) was a dropped or marked packet, then RE should be the highest non-dropped and non-marked packet before (LastAck + 1).

- If (LastAck + 1) was not a dropped or marked packet, the RE should be the greatest sequence number such that all data packets between (LastAck + 1) and RE, inclusive, were received and not ECN-marked. Clearly (RE >= LastAck + 1).

- LE should be the smallest sequence number such that all data packets between LE and RE, inclusive, were received and not ECN-marked. Clearly (LE <= RE).

The first bit of the final byte is the Nonce Echo. It equals the base-2 modulus of the number of received ECN Nonce packets between LE and RE, both included.

Note that the interval [LE, RE] would be the largest non-loss interval containing the first packet received since the last report, or, if that was a dropped packet, containing the run before this drop. That is, [LE, RE] would continue to grow during non-drop and non-mark periods.

7. Application Requirements

As described in the TFRC specifications [1], this CCID should not be used by applications that change sending rate by varying packet size, rather than varying the rate at which packets are sent.

8. Thanks

We thank Mark Handley for his help in defining CCID 3.

9. References

[1] M. Handley, J. Padhye, and S. Floyd. TCP Friendly Rate Control (TFRC): Protocol Specification. draft-ietf-tsvwg-tfrc-02.txt, work in progress.

[2] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. Proc ACM SIGCOMM 1998.

[3] E. Kohler, M. Handley, S. Floyd, and J. Padhye. Datagram Control Protocol. Work in progress.

[4] David Wetherall, David Ely, and Neil Spring. Robust ECN Signaling with Nonces. draft-ietf-tsvwg-tcp-nonce-00.txt, work in progress.

[5] S. Floyd, E. Kohler. Profile for DCP Congestion Control ID 2: TCP-like Congestion Control. Work in progress.

[6] K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. RFC 2481.

10. Authors' Addresses

Jitendra Padhye <padhye@aciri.org>
Sally Floyd <floyd@aciri.org>
Eddie Kohler <kohler@aciri.org>

AT&T Center for Internet Research at ICSI (ACIRI),
ICSI,
1947 Center Street, Suite 600
Berkeley, CA 94704.