Workgroup: ICCRG Working Group
Internet-Draft: draft-romo-iccrg-ccid5-00
Published: 25 October 2021
Intended Status: Experimental
Expires: 28 April 2022
Authors: N. Romo   J. Kim   M. Amend
DT   DT   DT

Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 5

Abstract

This document contains the profile for Congestion Control Identifier 5 (CCID 5), BBR-like Congestion Control, in the Datagram Congestion Control Protocol (DCCP). CCID 5 is meant to be used by senders who have a strong demand on low latency and require a steady throughput behavior.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document contains the profile for Congestion Control Identifier 5, BBR-like Congestion Control, in the Datagram Congestion Control Protocol (DCCP) [RFC4340]. DCCP uses Congestion Control Identifiers, or CCIDs, to specify the congestion control mechanism in use on a half-connection.

The BBR-like Congestion Control CCID5 sends data following the guidelines and principles of TCP BBR [I-D.cardwell-iccrg-bbr-congestion-control]. i.e, it estimates the path characteristics, to later update accordingly the sending data behavior. It achieves an optimal point of operation by keeping the amount of data in flight at the BDP (Bandwidth Delay Product) level, avoiding the abrupt Bandwidth changes typical of loss based congestion control algorithms.

2. Convention and Notation

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].

A DCCP half-connection consists of the application data sent by one endpoint and the corresponding acknowledgements sent by the other endpoint. The terms "HC-Sender" and "HC-Receiver" denote the endpoints sending application data and acknowledgements, respectively. Since CCIDs apply at the level of half-connections, we abbreviate HC-Sender to "sender" and HC-Receiver to "receiver" in this document. See [RFC4340] for more discussion

3. Usage

CCID5 congestion control algorithm is aimed to achieve a high bandwidth and low latency by the active probe of the end-to-end link capacity. The active probe helps hosts to adjust their sending rates before a packet loss happens at a buffer on the path. As a result, the communication path experiences a consistent and low latency by avoiding unnecessary packet drops at buffers.

Since CCID5 effectively avoids unnecessary packet losses, the spiky traffic behavior, that is commonly caused by traditional TCP congestion control mechanisms, is suppressed. This leads to a stable throughput throughout the connection period and thus yields a higher throughput than that with a loss-based congestion control mechanism.

Therefore, CCID5 suits applications that require consistent low latencies and stable high bandwidth. This includes multimedia streaming, online video gaming, video conferencing, and latency-sensitive industry applications such as industrial robots and autonomous vehicles are usage examples of CCID5.

3.1. Relationship with TCP BBR and CCID2

The CCID5 congestion control mechanism closely follows TCP's [I-D.cardwell-iccrg-bbr-congestion-control]|BBR congestion control algorithm, replicating the functions intended to estimate the path characteristics and to determine the pace and the amount of data to send. However, CCID5 must also comply with the DCCP requirements for a CCID profile ([RFC4340] Section 10.4) and define how the data is going to be acknowledged.

For this purpose, CCID5 implements the format of the ACK packets, the timing of their generation, and how they are congestion controlled. CCID5 uses the same ACK format as CCID2, including ACK vectors containing the same information that can be found in SACK options, and implements the ACK ratio as ACK congestion control mechanism.

In addition, the different variables and functions used to track packets in flight, packets acknowledged, and their corresponding sending and arrival times as well as the function to detect application-limited periods are replicated from the CCID2 implementation

3.2. Multiple-path communications

CCID 5 congestion control algorithm is adopted from TCP's BBR congestion control algorithm with a multiple-path communication as a representative use-case example. Multiple-path communications do not only target to maximize the link capacity, but also are aimed to improve the availability on critical situations such as a link failure. With that regard, MP-DCCP has been proposed. MP-DCCP extends capabilities of DCCP into multiple concurrent connections. A study [paper] has shown that CCID5 improves the overall bandwidth and the end-to-end latency compared to loss-based congestion control algorithms in an MP-DCCP enabled network. The study has also shown that the latency difference among multiple paths has an influence on the overall performance of the communication. A smaller gap among available paths leads to a higher aggregation performance of the link capacity. CCID5 is designed to provide a low and stable latency over each of the available paths and thus has a potential to improve the multi-path communication performance.

3.3. Half-Connection Example

This example shows the typical progress of a half-connection using CCID 5's BBR-like Congestion Control, not including connection initiation and termination. The example is informative, not normative.

1. The sender transmits DCCP-Data packets, each one of them identified with a sequence number. The sending behavior is governed by two control parameters: congestion window and pacing rate. The congestion window limits the amount of packets in flight and the pacing rate limits the sending rate.

2. The Acknowledgment mechanism replicates CCID2 specifications. Thus, The sender sends an Ack Ratio feature option specifying the number of data packets to be covered by an Ack packet from the receiver and consequently, the receiver sends a DCCP-Ack packet acknowledging the data packets for every Ack Ratio data packets transmitted by the sender.

3. The sender continues sending DCCP-Data packets. Upon receiving DCCP-Ack packets, the sender examines their Ack Vectors to learn about acknowledged and marked or dropped data packets. With the information of the acknowledged packets, it proceeds to estimate round-trip times (as TCP does) and the delivery rate, following the algorithm described in [I-D.cheng-iccrg-delivery-rate-estimation].

4. The sender uses the round-trip time and delivery rate estimations to calculate the round-trip propagation delay

(RTprop) and the bottleneck bandwidth (BtlBw) of the path, following the specifications in ([I-D.cardwell-iccrg-bbr-congestion-control] Section 4.1) The RTprop and BtlBw are then used to update the values of the congestion window and pacing rate.

5. As in CCID2, the sender responds to lost or marked DCCP-Ack packets by modifying the Ack Ratio sent to the receiver and acknowledges the receiver's acknowledgements at least once per congestion window.

4. Connection Establishment

The connection establishment is as specified in ([RFC4341] Section 4)

5. Congestion Control on Data Packets

CCID 5 is based on the BBR congestion control mechanisms described in [I-D.cardwell-iccrg-bbr-congestion-control]. The subsequent sections, present a general description of such mechanisms and discuss the considerations to be addressed when used within the DCCP protocol.

BBR proposes an algorithm based on the characterization of the network path made through the estimation of the Bottleneck Bandwidth (BtlBW) and the Round Trip propagation time (RTProp) defined respectively as the maximum delivered rate and minimum RTT seen by the sender. The algorithm aims to achieve an optimal point of operation by fulfilling two conditions

1. The amount of data inflight must be equal to the Bandwidth Delay Product (BDP), guaranteeing that buffers are not being filled and therefore avoiding long delay generation

2. The bottleneck packet arrival must match the BtlBw to ensure its full utilization.

To match those conditions, the sending data behavior is updated, by using three control variables: Congestion window (which limits the amount of data in flight), pacing rate, and send quantum (which limits the amount of aggregated packets in case of segmentation offload). The calculation of the control parameters uses as input the estimated values of BtlBW and RTprop along with two dynamic gain factors named pacing_gain and cwnd_gain.

The estimation of the path parameters Rtprop and BtlBw follow the guidelines and pseudo-code described in [I-D.cheng-iccrg-delivery-rate-estimation] and [I-D.cardwell-iccrg-bbr-congestion-control]

5.1. State machine

The way the control parameters are updated is governed by the BBR state machine Illustrated in Figure 1. In the initial Startup state, the sending rate will increase rapidly until the pipe is detected to be full. Afterwards, the data rate will be reduced so any possible queue can be drained, to finally enter into the ProbeBW state, where the amount of data in flight is slightly increased to probe for more possible bandwidth available. From any of these states, the algorithm can jump into the ProbeRTT phase. Here the data inflight is reduced to probe for lower RTTs. Each state defines specific values for two dynamic gains: cwnd_gain and pacing_gain, which will finally be used in the calculation of the aforementioned control variables.

```
                |
                V
       +---> Startup  ----+
       |        |         |
       |        V         |
       |      Drain   ----+
       |        |         |
       |        V         |
       +---> ProbeBW -----+
       |      ^    |      |
       |      |    |      |
       |      +----+      |
       |                  |
       +---- ProbeRTT <---+
```

Figure 1: BBR State machine

5.2. Response to Idle and Application-Limited Periods

6. Acknowledgements

The Acknowledgement format and its generation mechanism SHOULD follow the same specifications established for CCID2[RFC4341]. Thus, each Acknowledgment MUST contain an ACK vector defined with the format described in ([RFC4340] section 1.3) And its generation frequency will be controlled by the sender by using the ACK ratio feature.

7. Discussion

7.1. ProbeRTT phase transitions

The transition to and from the probeRTT phase MIGHT imply drastic changes of the congestion window, thus the synchronization of the

ACK ratio between and receiver SHOULD be handled carefully. When entering this phase at least one Packet MUST be sent with the new value of the ACK ratio before the reduction of the congestion window to 4 packets is executed, otherwise, the receiver MIGHT not be able to send ACK packets, preventing the sender from updating the measurement of the RTProp and BtlBW variables and remaining in this phase longer than required. Following a similar logic, before leaving the phase and restoring the congestion window value, at least one packet MUST be sent updating the ack ratio value, otherwise, the receiver MIGHT not be able to keep the pace to acknowledge the arriving packets, and the missing ACKs MIGHT trigger a RTO timeout.

In addition to the synchronization of the ACK ratio, the sender and receiver MUST keep synchronized the Sequence and Acknowledgment validity windows, as defined in ([RFC4340] section 7.5) This adds an additional constraint to the BBR algorithm when leaving the ProbeRTT phase, as at least one RTT is necessary for the sender to ensure the synchronization before restoring the congestion window value, causing again a longer duration of the probeRTT phase. Thus, it might be necessary to consider the possibility of restoring the congestion window even if this synchronization has not yet been confirmed by the arrival of the last Acknowledgement sent by the receiver.

8. IANA Considerations

9. Acknowledgment

10. Informative References

[I-D.cardwell-iccrg-bbr-congestion-control]
Cardwell, N., Cheng, Y., Yeganeh, S. H., and V. Jacobson, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-cardwell-iccrg-bbr-congestion-control-00, 3 July 2017, <https://www.ietf.org/archive/id/draft-cardwell-iccrg-bbr-congestion-control-00.txt>.

[I-D.cheng-iccrg-delivery-rate-estimation]
Cheng, Y., Cardwell, N., Yeganeh, S. H., and V. Jacobson, "Delivery Rate Estimation", Work in Progress, Internet-Draft, draft-cheng-iccrg-delivery-rate-estimation-00, 3 July 2017, <https://www.ietf.org/archive/id/draft-cheng-iccrg-delivery-rate-estimation-00.txt>.

[paper] Romo Moreno, N., Amend, M., Rakocevic, V., Kassler, A., and A. Brunstrom, "CCID5 An implementation of the BBR Congestion Control algorithm for DCCP and its impact over multi-path scenarios", DOI 10.1145/3472305.3472322, June 2021, <https://doi.org/10.1145/3472305.3472322>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[RFC4341] Floyd, S. and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", RFC 4341, DOI 10.17487/RFC4341, March 2006, <https://www.rfc-editor.org/info/rfc4341>.

Authors' Addresses

Nathalie Romo Moreno
Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany

Email: nathalie.romo-moreno@telekom.de

Juhoon Kim
Deutsche Telekom
Winterfeldtstr. 21
10781 Berlin
Germany

Email: j.kim@telekom.de

Markus Amend
Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany

Email: Markus.Amend@telekom.de