

tcpm  
Internet-Draft  
Intended status: Informational  
Expires: 18 April 2026

G. Zhao  
R. Li  
China Mobile  
15 October 2025

ACK Frequency Adjustment Strategy Based on Congestion Control Algorithms  
draft-zhao-tcpm-ack-freq-adjustment-00

Abstract

TCP Delayed ACK is a widely deployed mechanism that can effectively reduce protocol overhead in many scenarios. The TARR option has been defined, which allows a sender to request a specific ACK frequency from a receiver. However, there is no clear method for adjusting the ACK frequency during data transmission over a TCP connection. This document provides a method for adjusting the ACK frequency by considering the different phases of congestion control algorithms.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 18 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. ACK frequency Adjustment Strategy Based on Congestion Control Algorithms . . . . .	3
2.1. Adjusting ACK Frequency According to BBR Congestion Control Phases . . . . .	3
2.2. Adjusting ACK Frequency According to Reno-like Congestion Control Phases . . . . .	5
3. Experiment . . . . .	5
4. IANA Considerations . . . . .	5
5. Security Considerations . . . . .	5
6. Contributors . . . . .	5
7. Acknowledgements . . . . .	5
8. References . . . . .	5
8.1. Normative References . . . . .	5
8.2. Informative References . . . . .	6
Authors' Addresses . . . . .	6

## 1. Introduction

The TCP Delayed ACKs mechanism, specified in [RFC1122], aims to reduce protocol overhead. This design allows combining multiple segments and saves ACK packets under many traffic patterns, leading to its widespread deployment. A TCP connection has different requirements for ACK feedback frequency during different congestion control phases. Effectively adjusting the ACK frequency plays a crucial role in calculating RTT and improving congestion control performance.

TCP congestion control algorithms slide the congestion window upon receiving ACKs to regulate the sending rate. Currently, widely used congestion control algorithms include CUBIC and BBR[I-D.ietf-ccwg-bbr]. CUBIC is a typical RENO-type congestion control algorithm and is the default in operating systems like Linux and Windows. It uses a cubic function as the congestion window growth function during the congestion avoidance phase to improve network bandwidth utilization. RENO-type congestion control algorithms control the sending rate based on received ACK packets,

adjust the congestion window to regulate the sending rate, and use lost packets as congestion signals to perform network congestion control. Their characteristics include slow start, congestion avoidance, fast retransmit, and fast recovery.

The BBR congestion control algorithm primarily works by periodically probing the bottleneck bandwidth (bandwidth and delay) of the link and adjusting the congestion window size based on this information to achieve high bandwidth utilization and low transmission latency. BBR consists mainly of four phases: Startup, Drain, Probe Bandwidth, and Probe RTT.

Startup phase: The main task of this phase is to rapidly increase the sending rate until the link's bottleneck bandwidth is reached.

Drain phase: The main task of this phase is to drain the backlog of packets in the link to reduce network latency.

Probe Bandwidth phase: The main task of this phase is to probe the link's bottleneck bandwidth to determine the optimal sending rate.

Probe RTT phase: The main task of this phase is to probe the network's Round-Trip Time (RTT) to determine network stability.

The latest versions of the BBR algorithm have introduced new phases, including cruise, refill, up, and down, to improve the design of the Probe Bandwidth phase. These enhancements aim to improve the fairness of BBR flows when sharing network capacity with other traffic and also include improved handling of packet loss.

## 2. ACK frequency Adjustment Strategy Based on Congestion Control Algorithms

### 2.1. Adjusting ACK Frequency According to BBR Congestion Control Phases

#### 1. Startup Phase

In the Startup phase, the sender needs to rapidly increase its sending rate to quickly reach the link's bottleneck bandwidth. To accelerate the rate increase, the ACK feedback frequency should be set to a high value to ensure the sender quickly receives acknowledgments from the receiver. Set the ACK feedback frequency to provide an immediate ACK after each data packet is received, meaning the receiver sends an ACK immediately for every data packet.

#### 2. Drain Phase

In the Drain phase, the sender's main task is to drain the backlog of packets in the link to reduce network latency. Since frequent sending rate adjustments are not necessary in this phase, the ACK feedback frequency should be set to a lower value to reduce unnecessary computational and bandwidth resource consumption. The ACK frequency can be determined according to the following formula[TACK]:

$$f = \min\{bw/(L * MSS), \text{ 硫} / RTT\_min\} \text{ Formula (1)}$$

Where:

f is the current ACK feedback frequency.

bw is the available bandwidth of the current network link.

L is a constant, with a default value.

MSS is the Maximum Segment Size.

硫 is the number of packets sent per RTT\_min.

RTT\_min is the minimum delay measured on the path of the transmission connection, corresponding to the BBR.min\_rtt parameter measured by the BBR algorithm. This RTT value determines the minimum amount of time required for the connection to sustain transmission at the bottleneck bandwidth rate, thus determining the minimum amount of data required for the connection to achieve full utilization.

### 3. Probe Bandwidth Phase

In the Probe Bandwidth phase, the sender's main tasks are to probe the link's bottleneck bandwidth to determine the optimal sending rate and to send data steadily at the current rate.

BBRv1 Version: During the 1.25x speed-up phase, obtain one ACK per data packet and calculate the bottleneck bandwidth of the link. In other phases, calculate and adjust the ACK acquisition frequency according to Formula (1).

BBRv2, BBRv3 Versions: The corresponding bandwidth probing phase is the "up" phase. During this phase, obtain one ACK per data packet and calculate the bottleneck bandwidth of the link.

### 4. Probe RTT Phase

In the Probe RTT phase, the sender's main task is to probe the network's RTT to determine network stability. In the BBRv1 version, the sending rate or congestion window is reduced to 4 packets, and

one ACK is obtained per data packet. In BBRv2 and v3 versions, the sending rate or congestion window is reduced to half of its original size, and one ACK is obtained per data packet.

## 5. Other Phases

TBD.

### 2.2. Adjusting ACK Frequency According to Reno-like Congestion Control Phases

TBD.

## 3. Experiment

TBD.

## 4. IANA Considerations

TBD.

## 5. Security Considerations

TBD.

## 6. Contributors

The following people have substantially contributed to this document:

Zhiqiang Li  
lizhiqiangyjy@chinamobile.com

Hongwei Yang  
yanghongwei@chinamoblle.com

## 7. Acknowledgements

TBD.

## 8. References

### 8.1. Normative References

[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>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 8.2. Informative References

- [TACK] Li, T. and K. Zheng, "TACK: Improving Wireless Transport Performance by Taming Acknowledgments", Proceedings of the 2020 Conference of the ACM Special Interest Group on Data Communication (ACM SIGCOMM) , August 2020.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [I-D.ietf-ccwg-bbr] Cardwell, N., Swett, I., and J. Beshay, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-ietf-ccwg-bbr-03, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-ccwg-bbr-03>>.

## Authors' Addresses

Guangyu Zhao  
China Mobile  
No.32 XuanWuMen West Street  
Beijing  
100053  
China  
Email: [zhaoguangyu@chinamobile.com](mailto:zhaoguangyu@chinamobile.com)

Ruifeng Li  
China Mobile  
No.32 XuanWuMen West Street  
Beijing  
100053  
China  
Email: [liruifengyjy@chinamobile.com](mailto:liruifengyjy@chinamobile.com)