TCPM Working Group C. Gomez
Internet-Draft UPC
Intended status: Experimental J. Crowcroft
Expires: August 23, 2021 University of Cambridge
February 19, 2021

TCP ACK Rate Request Option
draft-gomez-tcpm-ack-rate-request-02

Abstract

TCP Delayed Acknowledgments (ACKs) is a widely deployed mechanism that allows reducing protocol overhead in many scenarios. However, Delayed ACKs may also contribute to suboptimal performance. When a relatively large congestion window (cwnd) can be used, less frequent ACKs may be desirable. On the other hand, in relatively small cwnd scenarios, eliciting an immediate ACK may avoid unnecessary delays that may be incurred by the Delayed ACKs mechanism. This document specifies the TCP ACK Rate Request (TARR) option. This option allows a sender to indicate the ACK rate to be used by a receiver, and it also allows to request immediate ACKs from a receiver.

Table of Contents

1. Introduction

Delayed Acknowledgments (ACKs) were specified for TCP with the aim to reduce protocol overhead [RFC1122]. With Delayed ACKs, a TCP delays sending an ACK by up to 500 ms (often 200 ms, with lower values in recent implementations such as ~50 ms also reported), and typically sends an ACK for at least every second segment received in a stream of full-sized segments. This allows combining several segments into a single one (e.g. the application layer response to an application layer data message, and the corresponding ACK), and also saves up to one of every two ACKs, under many traffic patterns (e.g. bulk transfers). The "SHOULD" requirement level for implementing Delayed ACKs in RFC 1122, along with its expected benefits, has led to a widespread deployment of this mechanism.

However, there exist scenarios where Delayed ACKs contribute to suboptimal performance. We next roughly classify such scenarios into two main categories, in terms of the congestion window (cwnd) size and the Maximum Segment Size (MSS) that would be used therein: i) "large" cwnd scenarios (i.e. cwnd >> MSS), and ii) "small" cwnd scenarios (e.g. cwnd up to ~MSS).

In "large" cwnd scenarios, increasing the number of data segments after which a receiver transmits an ACK beyond the typical one (i.e. 2 when Delayed ACKs are used) may provide significant benefits. One

example is mitigating performance limitations due to asymmetric path capacity (e.g. when the reverse path is significantly limited in comparison to the forward path) [RFC3449]. Another advantage is reducing the computational cost both at the sender and the receiver, and reducing network packet load, due to the lower number of ACKs involved.

In many "small" cwnd scenarios, a sender may want to request the receiver to acknowledge a data segment immediately (i.e. without the additional delay incurred by the Delayed ACKs mechanism). In high bit rate environments (e.g. data centers), a flow's fare share of the available Bandwidth Delay Product (BDP) may be in the order of one MSS, or even less. For an accordingly set cwnd value (e.g. cwnd up to MSS), Delayed ACKs would incur a delay that is several orders of magnitude greater than the RTT, severely degrading performance. Note that the Nagle algorithm may produce the same effect for some traffic patterns in the same type of environments [RFC8490]. In addition, when transactional data exchanges are performed over TCP, or when the cwnd size has been reduced, eliciting an immediate ACK from the receiver may avoid idle times and allow timely continuation of data transmission and/or cwnd growth, contributing to maintaining low latency.

Further "small" cwnd scenarios can be found in Internet of Things (IoT) environments. Many IoT devices exhibit significant memory constraints, such as only enough RAM for a send buffer size of 1 MSS. In that case, if the data segment does not elicit an application-layer response, the Delayed ACKs mechanism unnecessarily contributes a delay equal to the Delayed ACK timer to ACK transmission. The sender cannot transmit a new data segment until the ACK corresponding to the previous data segment is received and processed.

With the aim to provide a tool for performance improvement in both "large" and "small" cwnd scenarios, this document specifies the TCP ACK Rate request (TARR) option. This option allows a sender to indicate the ACK rate to be used by a receiver, and it also allows to request immediate ACKs from a receiver.

2. Conventions used in this document

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].

3. TCP ACK Rate Request Functionality

A TCP endpoint announces that it supports the TARR option by including the TARR option format format (with the appropriate Length value, see Section 4) in packets that have the SYN bit set.

The next two subsections define the sender and receiver behaviors for devices that support the TARR option, respectively.

3.1. Sender behavior

A TCP sender MUST NOT include the TARR option in TCP data segments to be sent if the TCP receiver does not support the TARR option.

A TCP sender MAY request a TARR-option-capable receiver to modify the ACK rate of the latter to one ACK every R full-sized data segments received from the sender. This request is performed by the sender by including the TARR option in the TCP header of a data segment. The TARR option carries the R value requested by the sender (see section 4). For the described purpose, the value of R MUST NOT be zero. The TARR option also carries the N field, which MUST be ignored when R is not set to zero.

When a TCP sender needs a data segment to be acknowledged immediately by a TARR-option-capable receiving TCP, the sender includes the TARR option in the TCP header of the data segment, with a value of R equal to zero. When R is set to zero, the N field of the same option indicates the number of subsequent data segments for which the sender also requests immediate ACKs.

A TCP sender MAY indicate that it has a reordering tolerance of R packets by setting the Ignore Order field of the TARR option to True (see Section 4).

3.2. Receiver behavior

A receiving TCP conforming to this specification MUST process a TARR option present in a received data segment.

When the TARR option of a received segment carries an R value different from zero, a TARR-option-capable receiving TCP MUST modify its ACK rate to one ACK every R full-sized received data segments from the sender, as long as packet reordering does not occur. When R is different from zero, the receiving TCP MUST ignore the N field of the TARR option.

A TARR-option-capable TCP that receives a TARR option with the Ignore Order (I) field set to True (see Section 4), MUST NOT send an ACK

after each reordered data segment. Instead, it MUST continue to send one ACK every R received data segments. Otherwise (i.e., Ignore Order = False), such a receiver will need to send an ACK after each reordered data segment received.

If a TARR-option-capable TCP receives a segment carrying the TARR option with R=0, the receiving TCP MUST send an ACK immediately, and it MUST also send an ACK immediately after each one of the N next consecutive segments to be received. N refers to the corresponding field in the TARR option of the received segment (see Section 4).

4. Option Format

The TARR option presents two different formats that can be identified by the corresponding format length. For packets that have the SYN bit set, the TARR option has the format shown in Fig. 1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Kind      |     Length    |              ExID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: TCP ACK Rate Request option format for packets that have the SYN bit set.

Kind: The Kind field value is TBD.

Length: The Length field value is 4 bytes.

ExID: The experiment ID field size is 2 bytes, and its value is 0x00AC.

For packets that do not have the SYN bit set, the TARR option has the format and content shown in Fig. 2.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     Kind      |     Length    |              ExID             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      R      |I|       N       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: TCP ACK Rate Request option format.

Kind: The Kind field value is TBD.

Length: The Length field value is 6 bytes.

ExID: The experiment ID field size is 2 bytes, and its value is 0x00AC.

R: The size of this field is 7 bits. If all bits of this field are set to 0, the field indicates a request by the sender for the receiver to trigger one or more ACKs immediately. Otherwise, the field carries the binary encoding of the number of full-sized segments received after which the receiver is requested by the sender to send an ACK.

Ignore Order (I): The size of this field is 1 bit. This field either has the value 1 ("True") or 0 ("False"). When this field is set to True, the receiver MUST NOT send an ACK after each reordered data segment. Instead, it MUST continue to send one ACK every R received data segments.

N: The size of this field is 1 byte. When R=0, the N field indicates the number of subsequent consecutive data segments to be sent for which immediate ACKs are requested by the sender.

5. IANA Considerations

This document specifies a new TCP option (TCP ACK Rate Request) that uses the shared experimental options format [RFC6994], with ExID in network-standard byte order.

The authors plan to request the allocation of ExID value 0x00AC for the TCP option specified in this document.

6. Security Considerations

The TARR option opens the door to new security threats. This section discusses such new threats, and suggests mitigation techniques.

An attacker might be able to impersonate a legitimate sender, and forge an apparently valid packet intended for the receiver, in order to intentionally communicate a bad R value to the latter with the aim to damage communication or device performance. For example, in a small cwnd scenario, using a too high R value may lead to exacerbated RTT increase and throughput decrease. In other scenarios, a too low R value may contribute to depleting the energy of a battery-operated receiver at a faster rate or may lead to increased network packet load.

While Transport Layer Security (TLS) [RFC8446] is strongly recommended for securing TCP-based communication, TLS does not protect TCP headers, and thus cannot protect the TARR option fields carried by a segment. One approach to address the problem is using network-layer protection, such as Internet Protocol Security (IPsec) [RFC4301].

7. Acknowledgments

Bob Briscoe, Jonathan Morton, Richard Scheffenegger, Neal Cardwell, Michael Tuexen, Yuchung Cheng, Matt Mathis, Jana Iyengar, Gorry Fairhurst, and Stuart Cheshire provided useful comments and input for this document. Jana Iyengar suggested including a field to allow a sender communicate its tolerance to reordering. Gorry Fairhurst suggested adding a mechanism to request a number of consecutive immediate ACKs.

Carles Gomez has been funded in part by the Spanish Government through project PID2019-106808RA-I00, and by Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya 2017 through grant SGR 376.

8. References

8.1. Normative References

[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>.

[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>.

[RFC6994] Touch, J., "Shared Use of Experimental TCP Options", RFC 6994, DOI 10.17487/RFC6994, August 2013, <https://www.rfc-editor.org/info/rfc6994>.

8.2. Informative References

[RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <https://www.rfc-editor.org/info/rfc3449>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8490] Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S., Lemon, T., and T. Pusateri, "DNS Stateful Operations", RFC 8490, DOI 10.17487/RFC8490, March 2019, <https://www.rfc-editor.org/info/rfc8490>.

Authors' Addresses

Carles Gomez
UPC
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

Jon Crowcroft
University of Cambridge
JJ Thomson Avenue
Cambridge, CB3 0FD
United Kingdom

Email: jon.crowcroft@cl.cam.ac.uk