Configuring TCP’s Initial Window
draft-you-tcpm-configuring-tcp-initial-window-00

Abstract

This document discusses that TCP’s initial congestion window is not a constant in different use cases. It proposes a flexible method to configure the initial window in order to keep up with the current network state.

Requirements Language

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].

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 http://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 April 22, 2015.

Copyright Notice

Copyright (c) 2014 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

Google proposes to increase TCP’s initial congestion window (InitCwnd) to at least ten segments (about 15KB) [RFC6928]. While for Taobao, the biggest online shopping mall in China, some public material from Taobao discloses that Taobao is using IW7 in their network instead of IW10. When the TCP’s InitCwnd is set to 7, they get the best end-user experience in Taobao’s experiments. In Google’s experiments, the InitCwnd is configured using the InitCwnd option in the IP route command. Furthermore, all front-end servers within a data center are configured with the same InitCwnd. However, as the network properties at geographically diverse locations differ, one global InitCwnd for all servers cannot optimize the TCP performance.

This document discusses that TCP’s initial congestion window is not a constant in different use cases. It proposes a flexible method to configure the initial window in order to keep up with the current network state.
2. Terminology

This section contains definitions of terms used in this document.

- **TCP**: Transmission Control Protocol
- **RTT**: Round-Trip Time
- **RTO**: Retransmission Timeout

3. Current TCP’s Initial Window Configuration Methods

The performance of initial TCP connection and congestion control is often affected by TCP parameters, such as initial congestion window, slow start threshold etc., which are usually default in systems. While with the global network access speeds growing, these default values set by systems may not be suitable for all the usages in current networks.

For example, Google believes a modest increase of InitCwnd to 10 is the best solution for the near-term deployment. Google’s experiments consist of enabling a larger initial congestion window on front-end servers in several data centers at geographically diverse locations. In their experiments, the front-end servers run Linux with a reasonably standards compliant TCP implementation (the congestion control algorithm used is TCP CUBIC), and the initial congestion window is configured using the initcwnd option in the ip route command, i.e., InitCwnd=10.

Another example is Taobao, the biggest online shopping mall in China. Some public material from Taobao discloses that Taobao is using IW7 in their network instead of IW10. In Taobao’s experiments, when the TCP’s InitCwnd is set to 7, they get the best end-user experience.

As the application scenarios for Google and Taobao are definitely different, the InitCwnd values for optimal performance are different too. So using one fixed InitCwnd value (i.e. 10) for all cases is not appropriate.

Current TCP’s InitCwnd is a global variable on a server or host, for example, a host is configured with the same initial congestion window for all the applications. Those parameters can be changed by modifying the regedit or kernel. However, they can’t be modified dynamically, nor can be based on different granularities, such as per TCP flow. If they can be adjusted according to peak and off-peak times of Internet, server capability, network bandwidth, number of users, etc., maybe the performance of TCP connections could be effectively improved. For example, when there is no network
congestion or server overloading, TCP initial window size could be set bigger. While during the peak time of Internet, e.g. "Double-11" shopping festival of Taobao, the window size could be set smaller in order to avoid unnecessary congestion.

4. Configuring TCP’s Initial Window

Different applications may require different transport performance to satisfy various service requirements. This document proposes that TCP’s initial window could be configured via TCP API (Application Programming Interface) based on different granularities such as TCP flow, as shown in Figure 1. By using this method, applications could flexibly customize their own TCP parameters according to their specific requirements.

```
+----------------+
|                |
|  Applications  |
+----------------+

+----------------+
|                |
|  Socket APIs   |
+----------------+

+----------------+
|                |
|  OS            |
|    +---------+  |
|    |TCP Stack|  |
|    +---------+  |
+----------------+
```

Figure 1: TCP’s Initial Window Configuration

TCP API needs to be extended to allow applications to configure TCP’s initial window per TCP flow granularity or others. A detailed example can be found in Appendix A.

By using socket APIs to configure TCP’s initial window, it is flexible for applications to customize appropriate TCP parameters according to service requirements and network availability, to achieve better user experience. Moreover, TCP’s initial window can be configured based on different granularities, e.g. TCP flow, which is a finer granularity to meet the service diversity.

Even if a system supports setting the initial window via the new API proposed, not all applications will take advantage of this, and so a system-wide default initial window still will apply to these
applications. For those applications don’t knowing how to pick an optimal value for the initial window, it is not recommended to used this function.

4.1. Factors affecting TCP’s Initial Window

This section discusses some factors that need to be considered when determining an appropriate TCP initial congestion window. Regarding how to find the proper InitCwnd, it is TBD.

4.1.1. Web Object Size

[RFC3390] stated that the main motivation for increasing the initial window to 4 KB was to speed up connections that only transmit a small amount of data, e.g., email and web. The majority of transfers back then were less than 4 KB and could be completed in a single RTT (Round-Trip Time).

However, nowadays, the size of the average web page of the top 1000 websites passed 1600K for the first time in July 2014. At the same time the number of objects in the average web page increased to 112 objects. Google proposed to increase TCP’s initial congestion window to at least ten segments (about 15KB) [RFC6928], while Taobao thinks 7 segments is optimal.

4.1.2. Bandwidth

For low bandwidth networks, such as GSM (Global System for Mobile Communication) and GPRS (General Packet Radio Service), injecting high-speed data into low-speed links leads to congestion or even collapse. In such case, small initial congestion window for TCP connections is relatively safe, e.g. 1 to 4, to prevent network congestion caused by a sudden influx of data into network. However, for networks with sufficient bandwidth capacity, the value of TCP initial congestion window could be set bigger, but we also need to consider other factors such as latency, packet loss, etc.

4.1.3. Latency

In high latency networks, the duration of slow start stage has a big impact on the whole TCP performance. A small initial congestion window usually leads to a long slow start stage, which may seriously decrease the TCP performance. Especially for short-lived services, e.g., HTTP Web transaction, most data would be transmitted at the low speed rate if the slow start stage is too long. For such kind of application, increasing InitCwnd enables transmission to be finished in fewer RTTs. This would shorten the duration of slow start stage and avoid RTO (Retransmission Timeout).
Our experiments show the relations between the initial congestion window (horizontal axis) and transmission time when sending 50k data (vertical axis) in a lab simulation environment. We compare the results using different initial congestion windows (from 1 to 10) under different latency (50ms, 100ms, 200ms, 300ms) when sending 50k data. As we can see, when the initial congestion window is bigger, the duration is smaller.

4.1.4. Packet Loss Rate

Packet loss is usually caused by network congestion due to insufficient bandwidth discussed in Section 4.1.3, or network device problems, e.g. not enough storage in routers. Increasing InitCwnd would lead to data stream burst into networks then would aggravate the packet loss. So in high congestion environment, TCP initial congestion window should be set relatively small, e.g. 2 or 4.

Our experiments show the relations between the initial congestion window (horizontal axis) and transmission time when sending 50k data with different packet loss rate (vertical axis) in a lab simulation environment. We compare the results using different initial congestion windows (from 1 to 10) under packet loss rate (0.10%, 0.20%, 0.40%, 0.60%, 0.80%) when sending 50k data with fixed latency=50ms. As we can see, when the initial congestion window is bigger, the duration is smaller.

4.1.5. Concurrent TCP connections

For applications with too many concurrent TCP connections, it is not suggested to set too large initial congestion window since too many network resources would be occupied in a very short time. It’s against the TCP fairness and also easily results in network congestion.

5. IANA Considerations

This document does not introduce any new IANA considerations.

6. Security considerations

This document introduces a new method to configure the initial congestion window for TCP connections. This method facilitates application developers to tune TCP for their benefits. But it also has the possibility that packet loss may caused by inappropriate setting. However, as RFC6928 says, it is unlikely to lead to a persistent state of network congestion or collapse. So it does not introduce any new security issues.
7. Acknowledgement

The authors would like to thank Yoshifumi Nishida and Wesley Eddy for their detailed review and comments.

8. Normative References


Appendix A. Example of Configuring TCP’s InitCwnd

This example illustrates how to set TCP initial congestion window parameter.

/* Server establishes listen socket */
int = listen_fd = socket( AF_INET, SOCK_STREAM, 0 );

/* Server creates accept socket to accept the data received from the listen socket from client */
int conn_fd = accept( listen_fd, ..., ... );

/* Receiving data and analyzing it (take HTTP as an example, the data may be the GET message from the client to request some resources */
read( conn_fd, buf, size );

/* e.g. Server calculates the initial cwnd based on the size of the resources */
int cwnd = (send_size+(mss-1))/mss;

/* MAX_INITCWND could be bigger than IW10 */
if (cwnd > MAX_INITCWND) cwnd = MAX_INITCWND;

setsockopt(conn_fd, IPPROTO_TCP, TCP_INITCWND, (void*)&cwnd, sizeof(cwnd));

In this example, TCP initial cwnd is a new extended parameter for socket API.
Authors’ Addresses

Jianjie You
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, 210012
China

Email: youjianjie@huawei.com

Rachel Huang
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, 210012
China

Email: rachel.huang@huawei.com