INTERNET-DRAFT
Intended Status: Informational
Expires: January 2, 2015

R. Huang
J. You
Huawei

July 1, 2014

# Traditional TCP Problem Statement
## draft-huang-tsvwg-tr-tcp-ps-00

Abstract

This draft discusses the problem statement and consideration for existing TCP.

Status of this Memo

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

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/1id-abstracts.html

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

Table of Contents

# 1 Introduction

The Transmission Control Protocol (TCP) is a reliable, ordered, congestion-controlled, byte stream transport layer protocol that is widely used on the Internet. But many issues keep coming out all the time in all kinds of environments when using TCP, and many TCP variants are created to handle these problems. Although these variant TCPs have achieved success in their respective target applications, designing a certain TCP variant that could perform gracefully in all environments is still a great challenge. For example, with the deployment of high speed and bandwidth wireless networks, e.g., LTE and WiMAX, real-time applications such as multimedia over HTTP/TCP may require TCP to handle both wireless connections and typical wired high BDP (Bandwidth-Delay Product) networks. In addition, different applications may require different parameters of TCP to satisfy their specific requirements. However, current TCP is not flexible enough for applications to customize their own solutions based on their different requirements.

This draft discusses the problem statement and consideration for existing TCP.

## 1.1 Relationship with TAPS

TAPS [TAPS] is proposed to identify and specify services provided by existing IETF transport protocols and congestion control mechanisms. TAPS will provide guidance on choosing among available mechanisms and protocols to obtain a given transport service. However, TAPS will not work on a specific protocol, such as TCP. The issues associated with the TCP protocol may not be discussed in TAPS. This document focuses on the TCP issues and may be as a reference for TAPS.

# 2 Terminology

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 RFC 2119 [RFC2119].

BDP Bandwidth-delay product refers to the product of a data link’s capacity and its end-to-end delay.

# 3 Problem Statement

## 3.1 TCP Variants

Network has been experiencing explosive growth in the last few decades, traditional TCP is struggling to meet the demands of today’s high bandwidth and low latency applications. With its focus on

reliability and successful delivery at the expense of efficiency, TCP has become a limiting factor in the request to utilize and extract more value from the increased bandwidth and enhanced processing capabilities that define today's physical Internet. In this section, some issues and their corresponding TCP variants are presented. From that, we can see that different scenarios require different TCP variants to guarantee the applications working well. But most of the TCP variants are not implemented in kernels or even standardized, which means applications have plenty of software work to do.

### 3.1.1 High-Speed Transmission

The daily and rapid spread of the broadband environment is led by FTTx. In these circumstances, core networks are starting to see a genuine transition from 10 Gbps TO 40 Gbps. For example, 24 GB of genomic data may need to be transferred from Beijing to California for scientific researches; Some ISPs may create large media files, between 15-30GB, which must be transferred and manipulated multiple times; The inter-DC backup may have up to 40TB/day content synchronized with DR (Disaster Recovery) site. High-speed transmissions are everywhere, but TCP is weak for these scenarios. The CC (Congestion Control) algorithms are sensitive to loss and delay as they reduce the transmission speed. The reliability bound with CC suppresses the new data sending, and the in-order transmission cannot fit the out-order scenarios well.

To solve the TCP issues in high-speed transmission scenario, many solutions have proposed. They can be classified into 2 types: Traditional approaches, which are just tuning the TCP; and aggressive approaches, which totally abandon TCP. Traditional approaches like Reno TCP [RFC3782], HSTCP [RFC3649], STCP, FAST TCP [FastTCP], produce their new CC algorithms. But they don't always run well. For example, Reno TCP has the limits of high loss rate, low increase and serious oscillation, which badly affect applications to properly use network bandwidth; FAST TCP may lead to sharp decline of network performance, or even collapse because of improper selection of balance point. Aggressive approaches like UDT [UDT], Fasp [FASP], JSCAPE, are all based on UDP and CC, which are not implemented on transport layer and not so easy to use.

### 3.1.2 Latency for Apps

Some applications, such as voice, networked games and interactive services, are more affected by latency rather than throughput. Excessive network latency will cause applications to spend a large amount of time waiting for responses from their remote sides, then the bandwidth may not be fully utilized, and performance will suffer.

TCP has 3-way handshakes occupying at least one RTT (Round-Trip Time) before sending content. Therefore, many applications don't run over TCP. TCP's InitCwnd and congestion control would also increase latency. The default initCwnd inside current kernel is 3 SMSS. Google suggests increasing InitCwnd to 10 SMSS recently. But, in our opinion, different applications may require different initCwnd values, and it's not so easy for applications to adjust it. TCP's in-order transmission is not friendly to multiplex and also unnecessary to some web applications due to HOL (Head-of-line Blocking). HTTP has suffered this issue for a long time and now Google is bringing new SPDY protocol to improve it.

Currently, there are some TCP variants proposed to get rid of the 3-way handshakes. T/TCP [RFC1644] and TFO (TCP fast open) [I.D-ietf-tcpm-fastopen] are two proposals of bypassing 3-way handshakes to let SYN packet carry data. But T/TCP is vulnerable to DOS attacks because the source address could be forged and SYN packets with data that the receiver had to accept could be sent. Three-way handshakes make it much less likely for an off-path attacker to be able to open large numbers of TCP connections, which exhausts resources on the receiver.

### 3.1.3 Data Center

Cloud computing has attracted widespread concern from industry. With the development of new Internet services and use of new technologies, significant changes and trends are taking place in data centers. It brings new challenges and problems to data center networks. Data center network traffic patterns are changing quickly from traditional "north-south" traffic to "east-west" traffic which results in a lot of one-to-many and many-to-many communication between the servers. TCP protocol is the wide-spread communication used among data center servers. However, TCP is originally designed for low bandwidth and low latency WAN, thus it's not quite suitable for data center network with high bandwidth and low latency. Therefore, 2 new issues emerge in data center network using TCP, i.e., TCP incast and TCP unfairness. TCP incast may be produced by MapReduce implementations, such as Hadoop. TCP unfairness is usually caused by different flows sharing the same link. In that case, big flows occupy most of the queues in switches resulting in high latency and bad performance to small flows, which contain relatively less data. IRTF is creating a research group to study the issues and solutions for TCP in data center.

Current way to alleviate the TCP pains in DC is to increase the queue in switches so that it could avoid dropping packets by incast, but at the expense of high queuing delay and high cost of switches. Data Center TCP (DCTCP) [DCTCP] is a TCP variant for data center networks. It achieves better throughput than TCP, reducing queuing delays and

congestive packet drops via Explicit Congestion Notification (ECN) to notify feedback to the hosts. However, DCTCP does not work well for deadline sensitive applications as deadlines of network flows are not regarded in the protocol and it is usually difficult to implement because of the modifications of end-hosts and switches.

#### 3.1.4 Wireless Network

As TCP was designed specifically for wired networks, where the packet loss was mainly caused by congestions. When applied in wireless network, the performance of traditional TCP is significantly affected by the channel errors, random losses and temporary link failures in wireless network.

Many approaches, both Transport Level Proposals and Link Level Proposals, have been proposed to improve the performance of TCP over networks with wireless links. TCP Westwood+ [TCP Westwood+], one of Transport Level Proposals, optimizes the control of cwnd (Congestion Window) and ssthresh (Slow Start Threshold) by estimating the available bandwidth, which solve the traditional TCP problem, that lost packets will result in decreasing bandwidth utilization, to a certain extend. Although Westwood+ [TCP Westwood+] is able to outperform standard TCP in a wide range of application scenarios, it still suffers in the presence of a large BDP, due to the low responsiveness after a packet loss. Another Transport Level Proposal, like Indirect TCP, splits the TCP connection into two, each of which applies independent, optimized flow and congestion control. But this method loses TCP’s end-to-end semantics and increases the overheads of the proxy. Link Level Proposals, such as Snoop protocol [Snoop], propose to deploy an agent at the base station and performing retransmission of lost segments based on duplicate TCP ACKs while retaining end-to-end semantics.

### 3.2 TCP Parameters

The performance of initial TCP connection and congestion control is often affected by TCP parameters, such as slow start threshold and maximum sending window size, which are usually default in systems. Obviously, these default values set by system may not be most suitable for all the usages in current networks. For example, Google has increased TCP’s initial congestion window, i.e., Initcwnd, to 10 for its searching service, which turns out to have a better performance; Taobao, the biggest online shopping mall in china, sets the TCP InitCwnd to 7 to get the best end-user experience.

Current TCP parameters which are all global parameters can be changed by modifying the regedit and kernel, and they can’t be changed dynamically or based on TCP flows. However, if they can be adjusted

according to peak and off-peak time of Internet, ability of servers, network bandwidth and the amount of users, won't it maximize the performance of transport? For example, when there is no network congestion, server overloading, TCP initial window size could be set bigger; While when meeting peak time of Internet, e.g. "Double-11" Shopping Festival in China, TCP initial window size could be set smaller to avoid unnecessary congestion. Another example is that setting different slow start cwnd for different services.

## 4 Security Considerations

TBD

## 5 IANA Considerations

There is no IANA action in this document.

## 6 Acknowledgments

The authors would like to thank Spencer Dawkins for giving valuable comments and suggestions.

## 7 References

### 7.1 Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3782] Floyd, S., Henderson, T., and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 3782, April 2004.

[RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", RFC 3649, December 2003.

[RFC1644] Braden, R., "T/TCP -- TCP Extensions for Transactions Functional Specification", RFC 1644, July 1994.

[TAPS] https://sites.google.com/site/transportprotocolservices/charter-proposal

[FastTCP] Nick, Barone; Jin, Cheng; Low, Steven H. and Hegde, Sanjay (2006). "FAST TCP: motivation, architecture, algorithms, performance"

[UDT] Y.Gu, R.L.Grossman, UDTv4:Improvements on Performance and

Usability[J]. Gridnets, 2008, 2(1):9-23

[FASP] http://asperasoft.com/technology/transport/fasp/

[I.D-ietf-tcpm-fastopen] Y. Cheng, "TCP Fast Open", draft-ietf-tcpm-fastopen-08, March 2014.

[DCTCP] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in Proc. of ACM SIGCOMM 2010, New Delhi, India, Aug. 2010

[TCP Westwood+] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control", ACM Comp. Comm. Rev.,vol. 34, pp. 25 - 38, April 2004

[I-TCP] A. Bakre and B.Badrinath, "I-TCP, Indirect TCP for Mobile Hosts," in 15th International Conference on Distributed Computing Systems (ICDCS), 1995.

[Snoop] http://nms.lcs.mit.edu/~hari/papers/snoop.html

## Authors' Addresses

Rachel Huang
Huawei Technologies Co., Ltd.
101 Software, Yuhua District

EMail: rachel.huang@huawei.com

Jianjie You
Huawei Technologies Co., Ltd.
101 Software, Yuhua District

EMail: youjianjie@huawei.com