INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: April 30, 2015

R.Sallantin
E.bouttier
CNES/TAS/TESA
C.Baudoin
F.Arnal
Thales Alenia Space
E.Dubois
CNES
E.Chaput
A.Beylot
IRIT

October 27, 2014

Safe increase of the TCP's Initial Window Using Initial Spreading
draft-sallantin-tcpm-initial-spreading-00

Abstract

This document proposes a new fast start-up mechanism to improve the short-lived TCP connections performance.

Initial Spreading allows to safely increase the Initial Window size in any cases, and notably in congested networks.

Merging the increase in the IW with the spacing of the segments belonging to the Initial Window (IW), Initial Spreading is a very simple mechanism that improves short-lived TCP flows performance and does not deteriorate long-lived TCP flows performance.

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."

Copyright and License Notice

Table of Contents

1 Introduction

The increase in the Initial Window size is a key topic that keeps the IETF community active since many years. In order to best fit to the evolution of the Internet tendencies, it is thus frequently proposed to enlarge the IW size.

Lately, [RFC6929] has therefore updated [RFC3390] and proposed an IW of 10 segments instead of 3. Several articles and studies have demonstrated that this small change would allow the transmission of 90% of the connections in one RTT [DR10]. If this is without any doubt the best way to deal with short-lived TCP flows in an un-congested environments, the consequences of the release of a large initial burst in a congested network is still an open question in the community.

In [SB14], we showed that enlarging the IW impacts the buffers and then deteriorates the individual connection in a congested environment. Correlations between the segments sent in a same burst are therefore responsible for major impairments when regarding the short-lived connections:

- a decrease of the probability to successfully transmit the entire window.
- an increase of the probability of successive segment losses.
- a significant reduction of the number of potential Duplicated Acknowledgements that are necessary to trigger fast loss recovery mechanisms and not wait for a Retransmission Time Out. Moreover, regarding the peculiar case of the connections that could be sent in one RTT (number of segments to be transmitted inferior or equal to the upper bound value of the TCP’s IW), experiments showed that the loss of one segment of the Initial burst could not be recovered using recovery mechanisms [SB14].

Initial Spreading has been designed to tackle previous burst issues and enable a safe increase in the Initial Window [SB13].

Initial Spreading uses the best of Pacing and Increase in the Initial Window [RFC6928] to enable the transmission of a large number of segments in the first RTT and ensure that each segment is received with a high independent probability.

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].

3 Initial Spreading mechanism

Initial Spreading[SB13] spaces out a number of segments inferior or equal to the permitted upper bound value of the TCP's IW (e.g; RFC 6928 [RFC6928] suggests to use 10 for this value) across the first RTT before letting the TCP algorithm continue conventionally:

(1) The RTT is measured during the SYN-SYN/ACK exchange.

(2) According to the RTT value, a Spreading Time (Tspreading) is computed (cf. section 5). Depending on the number of segments to be sent, until n segments are sent every Tspreading.

(3) After the transmission of the IW, the regular TCP algorithm is used.

Thus, bursts do not downgrade the transmission of short-lived connections, but continue to prevent an overload of the network in the case of long-lived connections.

4 Spreading Time design

4.1 Constraints

It has been observed that most of the savings enabled by Initial Spreading in congested environments comes from the independence of the segments sent during the first RTT. Indeed, experimentations [SB13] and analytical model [SB14] showed that Initial Spreading, by preventing the initial burst, enables each segment of the IW to have an independent loss probability. This reduces the latency variance and then, the average latency.

But, precautions should be taken not to be dependent of a false measurement of the initial RTT during the SYN-SYN/ACK exchange or to deteriorate the performance in un-congested network.

To be efficient, Initial Spreading should therefore take the best of several constraints:

- Tspreading MUST be bounded to not be dependent of the RTT measurement.
- Tspreading MUST be large enough for the losses to be un-correlated.

- o Tspreading SHOULD be the shortest possible to not add an un-necessary delay (notably in un-congested network).

- o Implementation MUST be light.

4.2 Burst impact on losses

It has been observed [SB14] that 2 segments are belonging to one burst if they do encounter the same bottleneck buffer state, and that the minimal spreading depends on the bottleneck throughput. Segments spread with Tspreading < BottleneckThroughput/MSS will face the same buffer state, and then will not be spread enough for the losses to be un-correlated.

As the bottleneck throughput is unknown and can not be known before the transmission of the Initial Window, Tspreading should be selected to offer the best performance whatever the throughput.

4.3 Tmax

Tmax is the upper bound value of Tspreading. It has two main purposes:

- o it enables Initial Spreading not to be dependent on the RTT measurement. This last introduces some uncertainty in the mechanism and increases the latency variance.

- o it reduces the mean latency.

Tmax’s choice results then in a trade-off.

A larger Tmax would enable the Initial Spreading to be efficient with lower bottleneck throughput (cf. section 4.2) in congested networks, but would decrease the benefits of a large Initial Window in un-congested network. On the opposite, a lower Tmax would reduce the additional delay in un-congested network but would decrease the benefits of Initial Spreading in congested network. In case Tspreading would not be large enough to insure a loss independence, Initial Spreading does not introduce additional delay but performs in a similar way than RFC6928.

The authors RECOMMEND the use of a Tmax equal to 2 ms. This value enable Initial Spreading to perform well in all cases:

o In case of short RTT (ie <20 ms), Tspreading is set to RTT/IW.
o In case of large RTT, Tspreading is set to Tmax. If the duration of the RTT is due to the delay and not to the congestion, then the additional delay would be low in comparison with the RTT duration.

Otherwise, if the large RTT is due to the congestion, our numerous experiments showed that whatever the considered Tmax, using Initial Spreading outperforms the regular performance of a large Initial Window without Initial Spreading.

4.4 Algorithm

Tspreading is computed as follows:

1. RTT/n is compared to Tmax, the maximal value of spreading, with n the permitted upper bound value of the TCP's IW.
2. If RTT/IW < Tmax,
   Tspreading = RTT/IW
3. If RTT/IW >= Tmax,
   Tspreading = Tmax

5 Implementation considerations

In this section, we discuss a number of aspects surrounding the Initial Spreading implementations.

5.1 Timers

High resolution timers MUST be used instead of Jiffy timers to implement the Initial Spreading.

Using a jiffy timer may therefore result in the transmission of new bursts and reduce Initial Spreading benefits: emissions of multiple TCP flows are synchronized via the Jiffies timer, so when m parallel flows are sent, a burst of m segments may be transmitted.

Finally, using HRTimer enables to keep the Initial Spreading algorithm simple (cf. section 4.4), and notably to not use a lower bound value for Tspreading.

5.2 Pacing in AQM

The authors RECOMMEND to apply the pacing in the Active Queue Management (AQM). For example, an implementation based on the new FQ/pacing would enable:

- to apply the Initial Spreading algorithm and select the optimal Tspreading for each flow, even in the case of multiple TCP flows.

- to not suffer from the TSO/GSO limitations.

- to reduce the overload in the TCP stack.

5.3 Delayed Ack

The use of Delayed Ack (Del Ack) does not downgrade Initial Spreading efficiency.

Regarding long-lived connections and notably TCP's steady state, the effects of Del Ack are lessened by new TCP's flavors (such as TCP Cubic or Compound TCP [HR08][TS06]) which tend to adapt their congestion algorithm to take into account whether the receiver uses the Del Ack option or not. In doing so, they can prevent the connection from being too slow, and still continue to reduce acknowledgments traffic. In the event of short-lived connections, the use of Del Ack does not modify the transmission of the IW. There is then no change in the burst propagation.

6 Open discussions

In this section, we introduce possible improvements for Initial Spreading and new perspectives.

6.1 Increasing the upper bound TCP's IW to more than 10 segments

[DR10] have shown that an IW of 10 segments enables to send more than 90% of the web objects in one RTT. So the authors recommend to use Initial Spreading as a complement to [RFC6928].

If the average size of the web objects continues to evolve, Initial Spreading can be used to raise the IW size. Simulations and experiments showed even better results with an IW equal to 12.

Thus, Initial Spreading paves the way for the use of a larger IW. Further studies are still needed to assess the impact of a higher IW on the network, notably in term of individual performance, fairness, friendliness and global performance.

6.2 Initial Spreading and LFN

The space community designed middleboxes to mitigate poor TCP performance for network with large RTT [FA11]. Proxy Enhancement Performance (PEP) are generally used in LFN and in particular in

satellite communication systems [RFC3135] and offer very good TCP performance.

Nevertheless, some recent studies have emphasized major impairments occasioned by the use of satellite-specific transport solutions, and notably TCP-PEPs, in a global context. The break of the end-to-end TCP semantic, which is required to isolate the satellite segment, is notably responsible for an increased complexity in case of mobility scenarios or security context. This strongly mitigates PEPs benefits and reopens the debate on their relevance[DC10].

Many researchers have outlined that new TCP releases perform well for long-lived TCP connections, even in satellite environment [SC12], but continue to suffer from very poor performance in case of short-lived TCP connections.

Initial Spreading enables to reduce the RTT consequences for short-lived TCP connections and could be an end-to-end alternative to PEP.

7 Security Considerations

The security considerations found in [RFC5681] apply to this document. No additional security problems have been identified with Initial Spreading at this time.

8 IANA Considerations

This document contains no IANA considerations.

9 References

9.1 Normative References

[RFC3390] A. Allman and S. Floyd, "Increasing tcp’s initial window," RFC 3390, IETF, Proposed Standard, 2002.

[RFC6928] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, "Increasing tcp’s initial window," RFC 6928, IETF, Experimental, Jan. 2013.

[DR10] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, and N. Sutin, "An Argument for Increasing TCP’s Initial Congestion Window," SIGCOMM Comput. Commun. Rev., vol. 40, no. 3, pp. 26-33, Jun. 2010.

[SB13] R.Sallantin, C.Baudoin, E.Chaput, F.Arnal, E.Dubois and A-L.Beylot, "Initial spreading: A fast Start-Up TCP mechanism," Local Computer Networks (LCN), 2013 IEEE 38th Conference on , vol., no., pp.492,499, 21-24 Oct. 2013

[SB14] R.Sallantin, C.Baudoin, E.Chaput, F.Arnal, E.Dubois and A-L.Beylot, "A TCP model for short-lived flows to validate initial spreading," Local Computer Networks (LCN), 2014 IEEE 39th Conference on , vol., no., pp.177,184, 8-11 Sept. 2014

[AH98] A. Allman, C. Hayes, and S. Ostermann, "An evaluation of TCP with Larger Initial Windows," ACM Computer Communication Review, 1998.

[AS00] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the performance of TCP pacing," in INFOCOM, vol. 3, mar 2000, pp. 1157-1165.

[RFC5532] T. Talpey, C. Juszczak, "Network File System (NFS) Remote Direct Memory Access (RDMA) Problem Statement," RFC 5532, IETF, Informational, May 2009.

9.2 Informative References

[SC12] R. Sallantin, E. Chaput, E. P. Dubois, C. Baudoin, F. Arnal, and A.-L.Beylot, "On the sustainability of PEPs for satellite Internet access," in ICSSC. AIAA, 2012.

[RFC3135] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, IETF, Informational, June 2001.

[DF10] E. Dubois, J. Fasson, C. Donny, and E. Chaput, "Enhancing tcp based communications in mobile satellite scenarios: Tcp peps issues and solutions," in Proc. 5th Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), pages 476-483, 2010.

[FA11] A. Fairhurst, G. Arjuna, H. Cruickshank, and C. Baudoin, "Transport challenges facing a next generation hybrid satellite internet," in International Journal of Satellite Communications and networking, 2011.

[HR08] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-

Speed TCP Variant," SIGOPS Oper. Syst. Rev., vol. 42, no. 5, pp. 64-74, Jul. 2008.

[LC09] R. Lacamera, D. Caini, C. Firrincieli, "Comparative performance evaluation of tcp variants on satellite environments," in ICC'09 Proceedings of the 2009 IEEE international conference on Communications, pages Pages 5161-5165, 2009.

[TS06] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "Compound TCP: A Scalable and TCP-friendly Congestion Control for High-speed Networks," in 4th International workshop on Protocols for Fast Long-Distance Networks (PFLDNet), 2006.

Authors' Addresses

Comments are solicited and should be addressed to the working group's mailing list at iccrg@irtf.org and/or the authors:

Renaud Sallantin
CNES/TAS/TESA
IRIT/ENSEEIHT 2, rue Charles Camichel BP 7122
31071 Toulouse Cedex 7
France
Phone: +33 6 48 07 86 44
Email: renaud.sallantin@gmail.com

Cedric Baudoin
Thales Alenia Space (TAS)
26 Avenue Jean Francois Champollion,
31100 Toulouse
France
Email: cedric.baudoin@thalesaleniaspace.com

Fabrice Arnal
Thales Alenia Space
Email: fabrice.arnal@thalesaleniaspace.com

Emmanuel Dubois
Centre National des Etudes Spatiales (CNES)
18 Avenue Edouard Belin

31400 Toulouse
France
Email: emmanuel.Dubois@cnes.Fr

Emmanuel Chaput
IRIT
IRIT / ENSEEIHT 2, rue Charles Camichel BP 7122
31071 Toulouse Cedex 7
France
Email: emmanuel.chaput@enseeiht.fr

Andre-Luc Beylot
IRIT
Email: andre-Luc.Beylot@enseeiht.fr

Tcpm Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2015

J. You
R. Huang
Huawei
R. Barik
University of Oslo
D. M. Divakaran
I2R, A*STAR

October 25, 2014

Configuring TCP's Initial Window
draft-you-tcpm-configuring-tcp-initial-window-02

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 28, 2015.

Copyright Notice

Table of Contents

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. 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. 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.

4.2. 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.

4.3. 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.

4.4. 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.

4.5. 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. Configuring TCP's Initial Window

This document proposes that TCP's initial window could be automatically configured based on the flow size whenever this size is known, as shown in Figure 1.

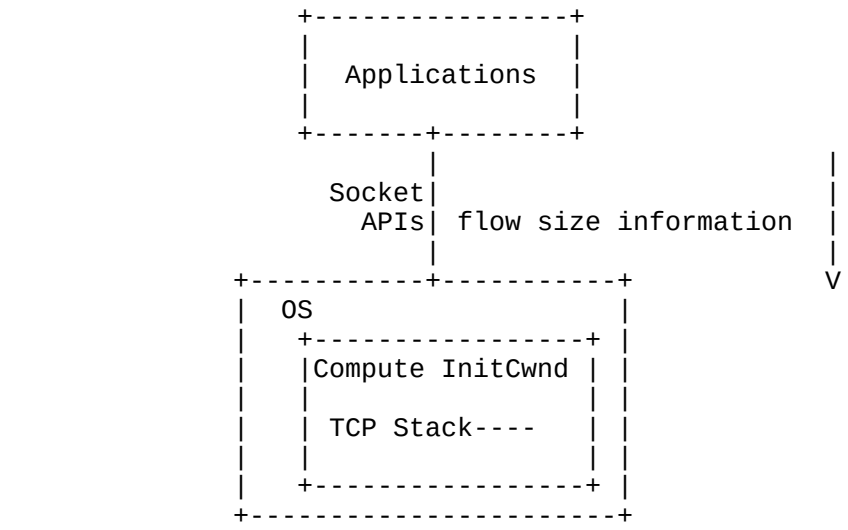

Figure 1: TCP's Initial Window Configuration

Intuitively, a function that determines InitCwnd of each flow based on its size can lead to better performance of small flows. In [WWIC][LCN][NoF] , this has been shown to be a benefit, irrespective of network conditions. The key idea is, the larger the flow-size, the smaller the InitCwnd. A lower bound for such an InitCwnd function can be the current standard for InitCwnd.

The results in [LCN] were derived using the following weighted function, where, for a flow of size s packets,

```
if s <= theta then
  InitCwnd = MaxIW
else
  InitCwnd = s/theta * MaxIW + (1-s/theta) * MinIW
end if
```

where theta is the flow size threshold used to distinguish between large flows and the rest, MinIW is the lower bound (four segments [RFC3390]), and MaxIW is the maximum InitCwnd that any connection can have (e.g. 10 [RFC6928]). The parameters, theta, MinIW and MaxIW are in number of TCP segments. Observe that, while small flows, defined by the threshold theta, will have InitCwnd as large as MaxIW, flows with size greater than theta will have InitCwnd closer to MinIW with increasing size.

This proposal assumes that flows will be able to know their sizes before the transfer begins. While this is true for many applications, for example, an HTTP query, a file transfer etc., there are also applications for which the flow-sizes can not be known in advance, for example, a streaming video. No changes are proposed for such flows.

6. IANA Considerations

This document does not introduce any new IANA considerations.

7. 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.

8. Acknowledgement

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

9. Normative References

[LCN] Barik, R. and Divakaran, D.M., "Evolution of TCP's initial window size", IEEE LCN 2013, Oct 2013.

[NoF] Barik, R. and Divakaran, D.M., "Development and Experimentation of TCP Initial Window Function", NoF 2014, Dec 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's Initial Window", RFC 3390, October 2002.

[RFC6928] Chu, J., Dukkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, April 2013.

[WWIC] Barik, R. and Divakaran, D.M., "TCP Initial Window: A Study", WWIC 2012, Jun 2012.

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

Runa Barik
University of Oslo
PO Box 1080 Blindern
Oslo N-0316
Norway

Email: runabk@ifi.uio.no

Dinil Mon Divakaran
I2R, A*STAR
1 Fusionopolis Way
#16-16 Connexix North Tower
Singapore 138632

Email: divakarand@i2r.a-star.edu.sg