Configuring TCP's Initial Window

draft-you-tcpm-configuring-tcp-initial-window-03

Jianjie You (youjianjie@huawei.com)
Rachel Huang (rachel.huang@huawei.com)
Runa Barik (runabk@ifi.uio.no)
Dinil Mon Divakaran (divakarand@i2r.a-star.edu.sg)

Presented by Michael Welzl

IETF 91
Honolulu, USA

November 11, 2014

Motivation

- The performance of initial TCP connection and congestion control is often affected by TCP parameters, such as initial congestion window (IW).
- Current TCP's IW is a global variable on a server or host.
- While with the global network access speeds growing, the IW set by systems may not be suitable for all the usages in current networks.
- For example, Google believes a modest increase of IW to 10 is the best solution for the near-term deployment, while Taobao, the biggest online shopping mall in China, is using IW7 in their network instead of IW10 for getting the best end-user experience.

Current TCP's IW Configuration Method

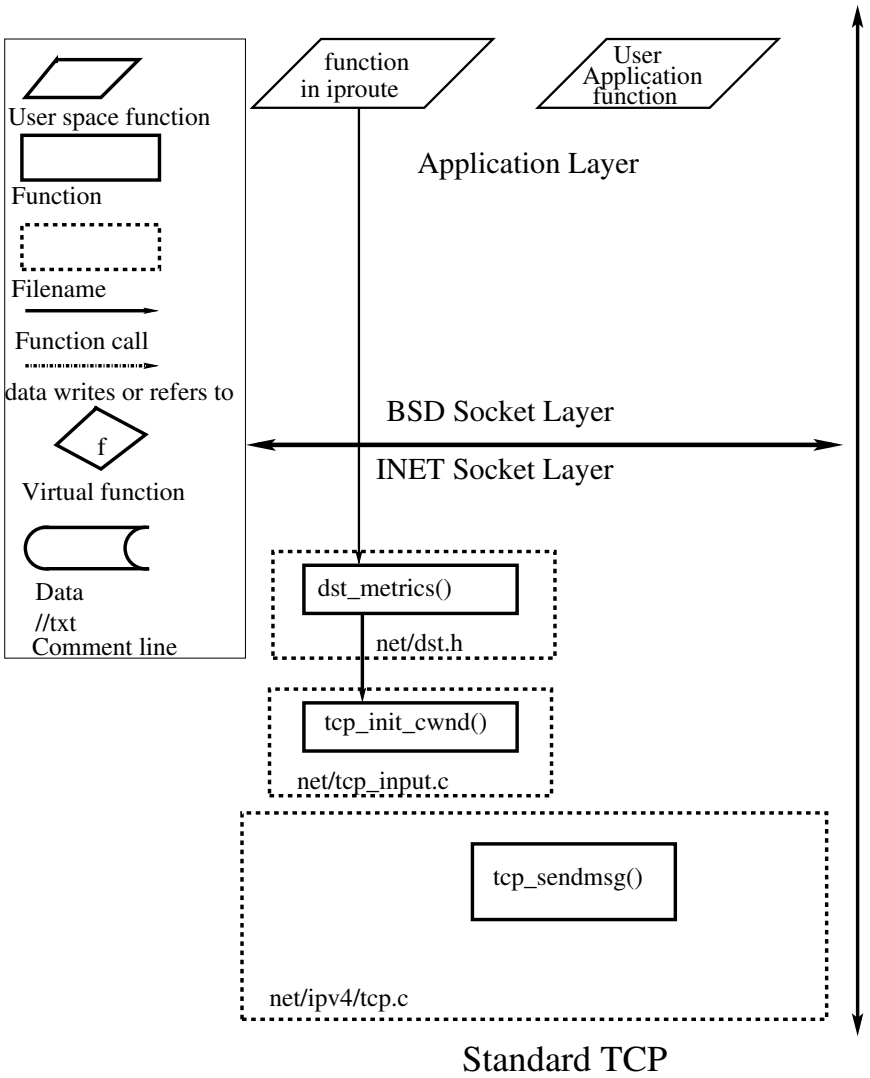

Figure: IW setting

The IW is configured:

- using the initcwnd option in the ip route command
- modifying regedit

Configuring TCP's IW

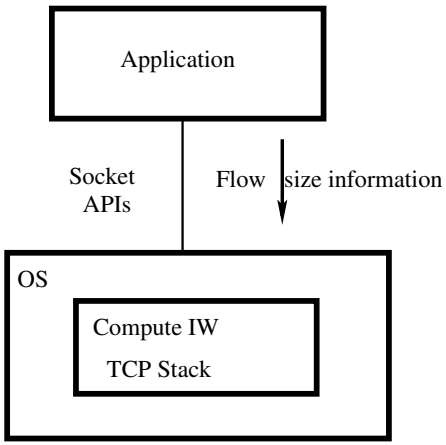

Figure: IW function setting

A function that determines IW of each flow based on its size, s, can lead to better performance of small flows, while IW almost not affecting the performance of large flows [1].

$$\begin{cases} V & s \leq \theta \\ \lfloor \frac{\theta}{s} \times V + (1 - \frac{\theta}{s}) \times MinIW \rfloor & s > \theta \end{cases}$$

θ denote the threshold to distinguish between large flows and the rest. V can be set to MaxIW (say 10; RFC 6928)

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

Issues

- APP Cheating
  - If an application tries to cheat the system by splitting a large flow into small ones, then it would have to weigh the benefits of being able to use a larger IW against the cost of the handshakes for closing and re-opening a connection, at least.
  - For large flows, staying in congestion avoidance (stable phase) is better than going to slow-start (transient phase) often, in the context of scheduling [2].
- How to find the correct value for θ, and V?

[2] S. Soudan, D. M. Divakaran, E. Altman, and P. Primet, "Equilibrium in size-based scheduling systems," in ASMTA '09

Initial Implementation and Experimentation

- Patch available on request, implemented in linux-3.7.4
- Testbed
  - NS2 simulations, and real experimentations
  - $\theta = 200$ KB, derived from $10,000$ flows from CAIDA data,

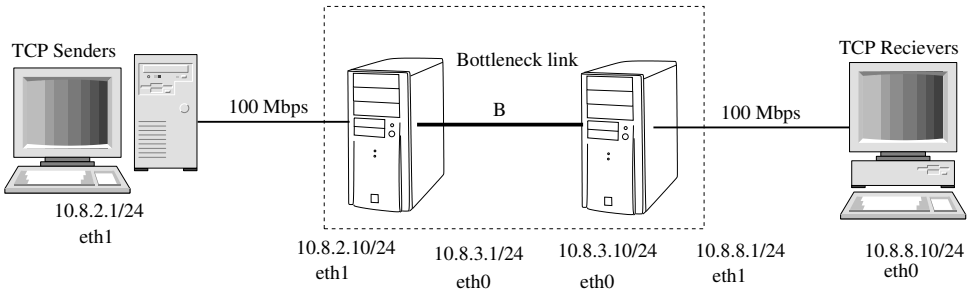

Figure: Network topology

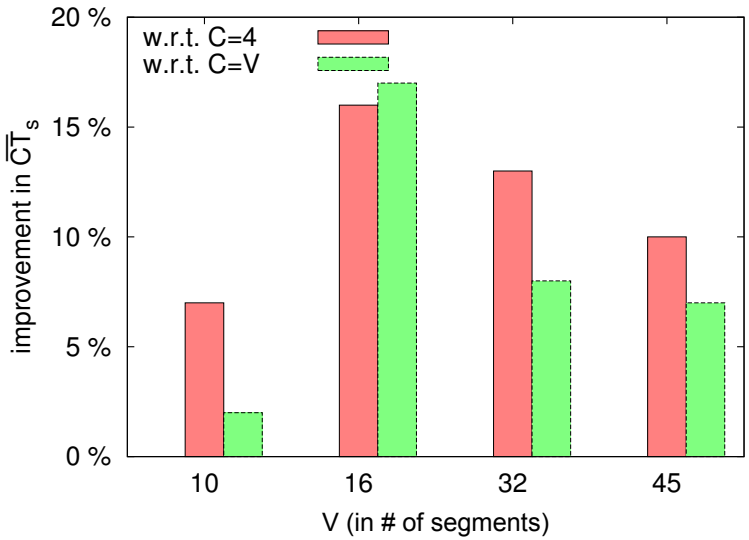

Figure: Percentage improvement in $\overline{CT_s}$ for IW function w.r.t. $C = V$ and $C = 4$

$\overline{CT_s}/\overline{CT_l}$: mean completion time for small/large flows.
C: constant IW

Results

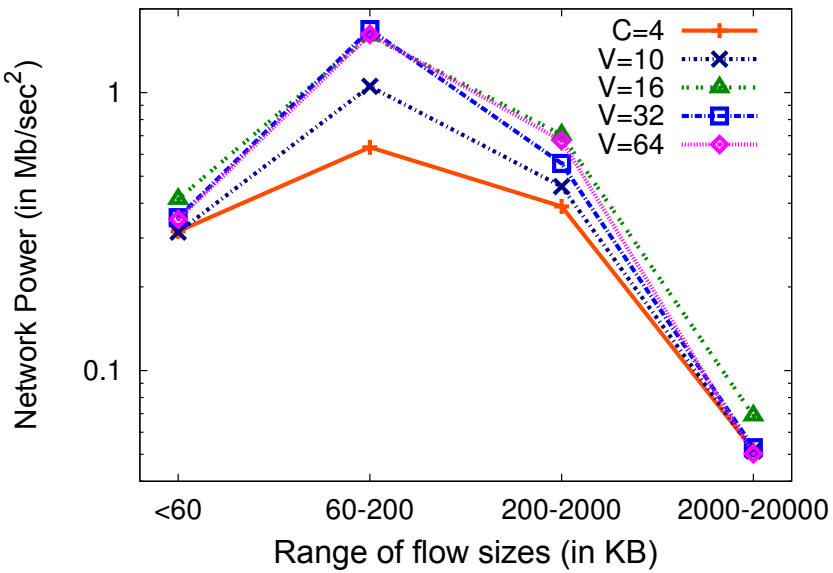

Figure: Network Power for ranges of flow-sizes (average_goodput/mean_completion_time)

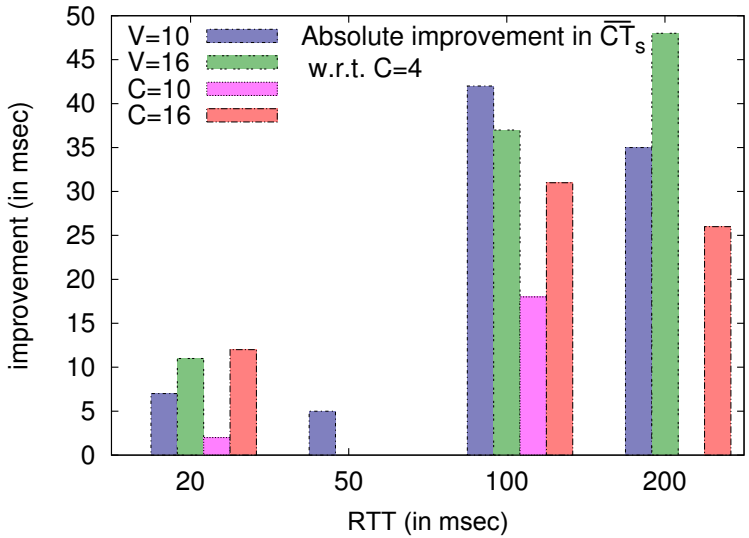

Figure: Percentage improvement in $\overline{CT_s}$ for IW function w.r.t. $C = V$ and $C = 4$ (with $B = 10$ Mbps, and varying RTT)

Results

Table: Comparison of other metrics

Parameter	RT_s	RR_s	RR_l	$\overline{CT_s}$	$\overline{CT_l}$
C=4	599	0.7	3.5	0.369	14.284
C=10	**893**	**0.8**	**4.0**	**0.351**	**14.243**
C=16	1659	1.4	5.7	0.375	13.78
C=32	1528	1.3	9.2	0.352	12.9
C=45	1270	1.2	11.2	0.359	11.77
V=10	805	0.8	4.0	0.343	14.148
V=16	**665**	**0.5**	**3.4**	**0.309**	**12.420**
V=32	754	0.6	5.2	0.323	13.845
V=45	898	0.8	7.5	0.334	12.6

RR_s/RR_l: Retransmission rate for small/large flows.
RT_s: # of time-outs encountered by small flows.

Next step

- solve the issues
- ask for the review and comments

Thanks!