Journal of Network and Computer Applications 41 (2014) 369–378

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A new high-performance TCP friendly congestion control over wireless networks

Satoshi Utsumi [a,b,*], Salahuddin Muhammad Salim Zabir [c,d]

[a] Tsuruoka National College of Technology, Department of Control and Information Systems Engineering, 104 Sawada, Inooka, Tsuruoka, Yamagata, Japan
[b] Tohoku University, Cyberscience Center, 6-3 Aoba, Aramaki, Aoba-ku, Sendai, Miyagi, Japan
[c] Orange Labs, Keio-Shinjuku Oiwake Bldg. 9F, 3-1-13 Shinjuku, Shinjuku-ku, Tokyo, Japan
[d] Waseda University, Graduate School of Global Information and Telecommunication Studies, 1-3-10 Nishiwaseda, Shinjuku-ku, Tokyo, Japan

ARTICLE INFO

Article history:
Received 19 June 2013
Received in revised form
8 February 2014
Accepted 11 February 2014
Available online 27 February 2014

Keywords:
Congestion control
Wireless link
Queueing delay
One-way delay
Timestamp option

ABSTRACT

In recent years, new delay-based congestion control mechanisms like Caia-Hamilton Delay (CHD) or Caia Delay-Gradient (CDG) are becoming popular due to their effectiveness even over wireless links. CHD and CDG are designed to behave friendly with conventional TCP NewReno flows. However, due to their delay-based window update mechanisms, CHD or CDG performance is often vulnerable to the co-existing aggressive TCP NewReno flows. In addition, in case of CHD, it is practically very difficult to choose the optimum values of tuning parameters with a view to ensuring the expected performance. In this paper, we therefore propose a new mechanism to overcome the above issues. We name it as Wireless Friendly Congestion Control (WFCC). Our objective is to devise a congestion control scheme that is (i) friendly with TCP NewReno flows over wireless links when deployed together, (ii) free from delicate operational parameters and (iii) robust against link errors under a wide range of network buffer space. We conduct a thorough evaluation of our proposed approach by simulation as well as emulation. Results show that WFCC (i) can lead up to 250% improvement in performance compared to TCP NewReno schemes and (ii) are friendly with TCP NewReno flows over wireless links.

1. Introduction

In recent years, mobile and wireless devices are being incorporated to the Internet at a rapid pace. This initiates the need for optimizing major applications to perform satisfactorily over wireless link. Wireless links are generally prone to a higher link error rate than their wired counterparts. Since conventional TCP protocols like TCP NewReno are designed to be used in wired networks with low link error rates, their original designs do not take link errors into account. Hence, they assume that segment losses occur solely due to congestion in networks (Jacobson, 1988). Whenever a segment loss occurs, conventional TCP infers congestion and the TCP sender reduces its sending rate as a remedial measure. Therefore, when conventional TCPs are used over wireless networks, they interpret all the losses to be originating from congestion and frequently lower their transmission rates unnecessarily. As such, deployment of conventional TCP over wireless links results in a decreased throughput (Balakrishnan et al., 1997).

Various end-to-end congestion control mechanisms have so far been proposed to handle this problem. Most of these mechanisms are reactive. That is, whenever a segment loss occurs, they attempt to figure out whether the actual reason for the loss was network congestion or wireless link error. We therefore mention them as 'loss based' schemes in this paper. In addition, a comparatively new 'proactive' approach for TCP congestion control has been flourishing in recent years. They aim at decoupling TCP performance from the network buffer space by using delay in the network as the parameter. We call them 'delay-based' approaches. Among these delay-based approaches, mechanisms like Caia-Hamilton Delay (CHD, Hayes and Armitage, 2010) and Caia Delay-Gradient (CDG, Hayes and Armitage, 2011) offer effective solutions for wireless link loss.

CHD and CDG are designed to behave friendly with conventional TCP NewReno flows. However, due to their delay-based window update mechanisms, CHD or CDG performance is often vulnerable to the co-existing aggressive TCP NewReno flows that ultimately occupy most of the bottleneck link bandwidth. In addition, delay based mechanisms are highly sensitive to the tuning of corresponding parameters. For example, in case of CHD, it is not possible to determine the most accurate value of the queuing threshold intuitively.

Previously, we proposed very TCP friendly congestion controls over wireless links, named as Utilization-based Congestion Control (UCC, Utsumi and Zabir, 2012). UCC provides a solution based on

* Corresponding author.
E-mail address: u-satoshi@tsuruoka-nct.ac.jp (S. Utsumi).

http://dx.doi.org/10.1016/j.jnca.2014.02.003
1084-8045

the utilization at the bottleneck link. We estimate the utilization on the bottleneck link using queueing theory and apply this for end-to-end congestion control.

In this paper, we propose a novel congestion control mechanism to overcome the above issues. We name this as Wireless Friendly Congestion Control (WFCC). Our proposed mechanism, WFCC, is based on the queueing delay at the bottleneck link and is (i) friendly with TCP NewReno flows over wireless links when deployed together, (ii) free from delicate operational parameters and (iii) robust against link errors under a wide range of network buffer space.

We conduct a thorough evaluation of our proposed approach by emulation. Results show that (i) WFCC can yield a performance improvement of 170% or more compared to TCP NewReno schemes and a performance improvement of 93% or more compared to UCC schemes over wireless links and (ii) it is friendly with TCP NewReno flows over wireless links.

The rest of this paper is organized as follows. Section 2 summarizes the most recent relevant research works. In Section 3 we present how one way delay can be utilized for deriving our other proposed mechanism, Wireless Friendly Congestion Control (WFCC). In Section 4, we discuss in detail how we have evaluated WFCC through emulation. Finally we conclude in Section 5.

2. Related works

A number of interesting loss based congestion control mechanisms have been proposed to date (Ludwig and Katz, 2000; Akyildiz et al., 2001, 2002). Since these are already quite well established mechanisms, we omit mentioning them in this section. On the contrary, congestion control mechanisms using delay parameters are relatively new and getting popular in recent years. Hence, we focus on them in this section. Table 1 summarizes some congestion controls using delay parameters (Hayes and Armitage, 2010).

2.1. Hamilton Delay (HD)-based congestion control

Leith et al. (2007) describe the case for delay-based Additive Increase Multiplicative Decrease (AIMD) congestion control, which provides end-to-end control with high utilization, low delay and zero congestion related packet loss. This idea was improved by Budzisz et al. (2009) for fair coexistence with loss-based TCP algorithms. We use the same notions as (Hayes and Armitage, 2010) to describe the Hamilton Delay-based congestion control (HD) algorithm.

On receipt of every ACK, cwnd(w) is evaluated as follows:

$$w_{i+1} = \begin{cases} \dfrac{w_i}{2}, & X < g(q_i) \\ w_i + \dfrac{1}{w_i} & \text{otherwise} \end{cases} \tag{1}$$

where $g(q_i)$ is the backoff probability function shown in Fig. 1 (as reported in Budzisz et al., 2009), $X \in [0, 1]$ is a random number, p_{max} is the maximum probability of backoff, $q_{max} = RTT_{max} - RTT_{min}$ is an estimate of the maximum observed queuing delay, q_{min} is a target minimum queuing delay, and q_{th} is a threshold that divides regions A and B.

When loss-based flows are on the link, the queue is pushed into region B. The delay-based flows have a lower probability of backoff in this region, enabling them to receive a fairer share of the available bandwidth. When loss-based flows are no longer on the

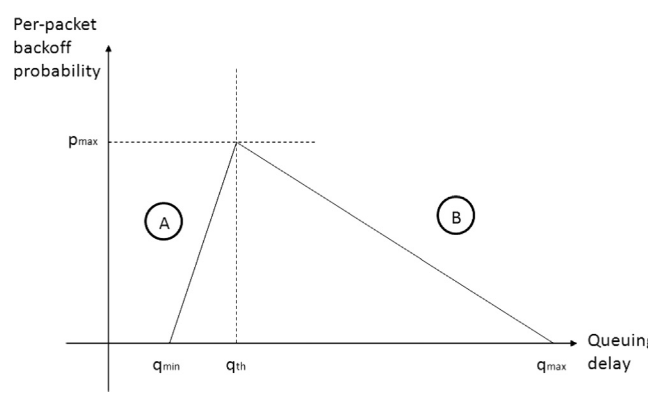

Fig. 1. Per-packet backoff probability function (Budzisz et al., 2009).

Table 1
Overview of congestion controls using delay parameters (Hayes and Armitage, 2010), where β is the multiplicative decrease factor, θ represents a delay threshold, τ_i is the ith RTT, τ_{min} is the smallest RTT, τ_{max} is the largest RTT, d_i is the ith one way delay).

Congestion control scheme	Delay measurements	Congestion inference	Congestion control
Congestion avoidance using Round-trip Delay (CARD, Jain, 1989)	RTT	Normalized delay gradient, $\left(\frac{\tau_i - \tau_{i-1}}{\tau_i + \tau_{i-1}}\right) > 0$	AIMD $(\beta = \frac{7}{8})$
DUAL (Wang and Crowcroft, 1992)	Every 2nd RTT	$\tau_i > \frac{\tau_{min} + \tau_{max}}{2}$	AIMD $(\beta = \frac{7}{8})$
TCP Vegas (Brakmo and Peterson, 1995)	RTT	$\tau_i > \tau_{min} + \theta$	AIAD
Fast TCP (Jin et al., 2004; Wei et al., 2006)	Smoothed RTT	Similar to Vegas	MIMD
TCP-LP (Kuzmanovic and Knightly, 2006)	Smoothed one way delay	$\overline{d_i} > d_{min} + \delta(d_{max} - d_{min})$	AIMD
TCP-Africa (King et al., 2005)	Smoothed RTT	Similar to Vegas	Dual mode
Compound TCP (CTCP, Tan et al., 2006)	Smoothed RTT	Similar to Vegas	Dual mode
Probabilistic Early Response TCP (PERT, Bhandarkar et al., 2007)	RTT and smoothed RTT	Threshold based on queuing delay $(q_j = \tau_j - \tau_{min})$	Probabilistic backoff
Hamilton Delay (HD, Leith et al., 2007; Budzisz et al., 2009)	RTT	Threshold based on queuing delay	Probabilistic backoff
Caia-Hamilton Delay (CHD, Hayes and Armitage, 2010)	RTT	Threshold based on queuing delay	Probabilistic backoff
Caia Delay-Gradient (CDG, Hayes and Armitage, 2011)	RTT	Type of RTT gradient	Probabilistic backoff
Utilization-based Congestion Control (UCC, Utsumi and Zabir, 2012)	Utilization	Threshold based on link utilization	AIMD