Improving the Performance of Multipath Congestion Control over Wireless Networks

Tuan-Anh Le
Department of Information Technology
Thudaumot University
Binh Duong, Vietnam
Email: letuanh@tdmu.edu.vn

Abstract— **Recently, portable devices with wireless network interfaces are becoming more popular than desktop PCs. While multipath coupled congestion control with ”linked increases” algorithm (MPTCP-LIA) and its advanced ”opportunistic linked increases” algorithm (MPTCP-OLIA) were designed for wired networks. As a result, their performance is degraded significantly in wireless environments, where packet losses often are caused by random error rather than by network congestion as in wired networks. In this work, we analyze and evaluate several existing multipath congestion control algorithms, and then propose a design of multipath TCP Veno (MPVeno) for wireless networks. Our simulation results show that MPVeno can achieve better goodput than MPTCP-OLIA over wireless links while preserving fair competition with MPTCP-OLIA in wired networks.**

I. Introduction

Nowadays, portable devices, including smart-phones, tablets, and laptops, are becoming more popular than desktop PCs. Meanwhile, their Internet access through wireless interfaces is on the rise. Multipath TCP, which allows packets of a connection to be sent simultaneously across multiple different paths, has been approved by the IETF (RFC 6824) [1]. The benefits of such end-to-end multipath transport using both 3G and WiFi networks are higher throughput, smoother handoff, and increased battery life. We argue that coupled congestion control with ”linked increases” algorithm (named MPTCP-LIA) (RF6356) [2] or its advanced ”opportunistic linked increases” algorithm (MPTCP-OLIA) for multipath TCP is not a one-size-fits-all for the different applications (e.g., data vs. multimedia streams) and environments (e.g., high-speed and/or long delay networks, and wireless networks). The goal of this paper is to improve the performance of MPTCP-LIA and MPTCP-OLIA, which is motivated by the following problem: The performance of MPTCP-LIA and MPTCP-OLIA is degraded significantly over wireless links, where packet losses often are caused by random error in wireless links rather than by network congestion.

Therefore, finding an efficient solution for multipath transmission over wireless links is one of the most challenging issues. Although our previous multipath TCP Westwood (named MPTCPW) [3] can outperform MPTCP-LIA over wireless links, but MPTCPW cannot fairly share bandwidth with existing MPTCP-LIA and MPTCP-OLIA flows in wired networks.

Recently, MPTCP-OLIA (Opportunistic Linked Increases Algorithm) [4] was proposed to overcome non-Pareto optimality of MPTCP-LIA. Any multipath congestion control algorithm must meet the three goals [2]: (i) Do not harm: A multipath congestion control algorithm should fairly share bandwidth with existing single-path TCP and MPTCP-LIA protocol; (ii) Performance improvement: Its total throughput should be at least that of single-path TCP flow on the best path that for it; (iii) Congestion balance: It can balance end-to-end congestion between paths.

In this paper, firstly, we evaluate some existing multipath congestion control algorithms for wireless networks. Secondly, we introduce a multipath transport version of single-path conventional TCP Veno [5], which can improve the TCP performance over wireless links.

The rest of this paper is organized as follows: We summarize the previous works relevant to multipath congestion control algorithms over wireless networks and analyze them in Section II. We describe a design of multipath TCP Veno over wireless networks in Section III. And we evaluate and compare our proposal with several existing multipath protocols in terms of fairness, goodput over wireless networks, and friendliness to MPTCP-OLIA in wired networks in Section IV. Finally, we conclude our work in Section V.

II. Multipath Congestion Control Algorithms over Wireless Networks

In this section, we summarize several multipath congestion control algorithms has been proposed for wireless networks. They can be classified into two approaches: Uncoupled congestion control approach, and Coupled congestion control approach. As we known, a multipath TCP connection comprises of two or more sub-flows, which perform congestion control functions on their paths.

For uncoupled congestion control approach, congestion control just allow to simultaneously spread data across multiple paths without any coordination of congestion control between paths. For example, Wireless multipath SCTP (WiMP-SCTP) [6] switches between data-striping and data-duplicating modes according to calculation of the amount of transmission errors of all paths used. In the data-striping mode, WiMP-SCTP improves the throughput by using independent congestion control between wireless links. Westwood SCTP [7] simultaneously sends data on paths. Each sub-flow in Westwood

978-1-4799-1089-2/13/$31.00 ©2013 IEEE

SCTP estimates the available bandwidth on its path, and performs congestion control independently. As a result, both WiMP-SCTP and Westwood SCTP fail in meeting the above mentioned goals. Therefore, we do not evaluate the algorithms in this approach.

By contrast, in coupled congestion control approach, a sub-flow increases its congestion window size according to not only network condition of the path that for it, but also network condition of other paths. Our proposal falls into this approach. We review and evaluate coupled congestion control algorithms over wireless links in the next section.

A. MPTCP-Linked Increases Algorithm (MPTCP-LIA)

MPTCP-LIA [2] uses "linked increases" to couple congestion control between paths. Sub-flow on path r updates its congestion window size w_r as the following rules.[1]

$$\text{Increase: } w_r \leftarrow w_r + \min\left(\frac{a^{2-\phi} w_s^{1-\phi}}{w_{sum}^{2-\phi}}, \frac{1}{w_r}\right),$$
$$\text{Decrease: } w_r \leftarrow w_r - w_r/2,$$

where

$$a = w_{sum} \frac{\max(w_r/rtt_r{}^2)^{1/(2-\phi)}}{(\sum_i x_i)^{2/(2-\phi)}},$$

$w_{sum} = \sum_i w_i$, data rate $x_i = w_i/rtt_i$, and parameter $\phi \in [0, 2]$ generates a family of MPTCP-LIA protocols and a trade-off between flappiness of path usage and congestion balance. a determines the responsiveness of the protocol. With $\phi = 2$, MPTCP-LIA becomes uncoupled congestion control algorithm. With $\phi = 0$, MPTCP-LIA can achieve perfect congestion balance, but it is flappy to simultaneously obtain available bandwidth on paths even all paths were experienced the same congestion level [8]. To balance the trade-off, regular MPTCP-LIA (RFC6356) uses $\phi = 1$. However, through experiments and analysis, the authors in literature [4] have proved that MPTCP-LIA is non-Pareto optimality, which indicates that MPTCP-LIA can reduce the throughput of other single-path users. Therefore, MPTCP-OLIA has been proposed to overcome this problem [4].

B. MPTCP - Opportunistic Linked Increases Algorithm (MPTCP-OLIA)

MPTCP-OLIA [4] modifies Increase and Decrease rules on path r as follows.

$$\text{Increase: } w_r \leftarrow w_r + a_r/w_r + \alpha_r/w_r, \quad (1)$$
$$\text{Decrease: } w_r \leftarrow w_r - w_r/2,$$

where

$$a_r = x_r^2 / \left(\sum_i x_i\right)^2, \quad (2)$$

[1]The sender carries out Increase rule whenever receiving a positive ACK, and Decrease rule upon detecting a packet loss.

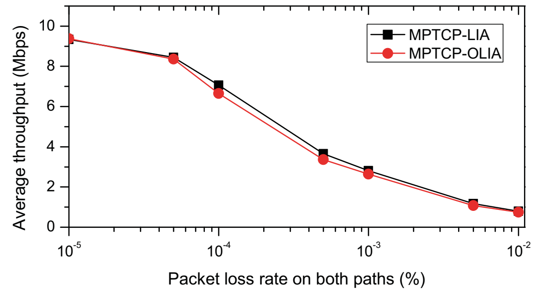

Fig. 1. The average throughput of two-path MPTCP-LIA and MPTCP-OLIA vs. varying packet losses on paths 1 and 2.

$$\alpha_r = \begin{cases} \frac{1}{N \times |\mathcal{H} \backslash \mathcal{L}|} & \text{, if } r \in \mathcal{H} \backslash \mathcal{L} \neq \emptyset \\ \frac{-1}{N \times |\mathcal{L}|} & \text{, if } r \in \mathcal{L} \text{ and } \mathcal{H} \backslash \mathcal{L} \neq \emptyset \\ 0 & \text{, otherwise.} \end{cases} \quad (3)$$

$\mathcal{L}$ is the set of the sub-flows that has the largest congestion window size at the time observed. $\mathcal{H}$ is the set of the sub-flows that has the highest throughput (or the best paths) at the time observed. N is the number of sub-flows established during a MPTCP-OLIA connection. $|.|$ denotes the cardinality of a set. N denotes the number of paths used in a multipath TCP connection.

When we omit term α_r/w_r in Increase rule in Eq. (1), MPTCP-OLIA will be equivalent to MPTCP-LIA with $\phi = 2$, where both algorithms are flappy in their path usage.

Now, we evaluate both MPTCP-LIA and MPTCP-OLIA in terms of throughput over wireless links by using simulation. Fig. 1 shows that both algorithms are degraded their throughput as random packet loss rate on wireless links increases because they have the similar behavior to packet loss.

C. Multipath TCP Westwood Algorithms

In this section, we present two existing multipath TCP Westwood designs, and then compare their throughput performance over wireless links.

Weighted multipath TCP Westwood (wMPTCPW) [9] adjusts the weighted parameter in the window increase rule according to the square of ratio of the maximum estimated bandwidth among paths to the total estimated bandwidth of all paths. Congestion window update rules on path r are

$$\text{Increase: } w_r \leftarrow w_r + \theta/w_r,$$
$$\text{Decrease: } w_r \leftarrow B_r \times RTT_{min,r},$$

where

$$\theta = \max_s(B_s^2) / \left(\sum_i B_i\right)^2,$$

B_r is the estimated available bandwidth on path r, [2] $RTT_{min,r}$ denotes the minimum RTT during sub-flow connection on path r.

[2]The sender estimates available bandwidth B by measuring inter-arrival time of ACK packets, and then filtering out the noise by low pass filter [10].

Other multipath TCP Westwood version [3] (MPTCPW) was designed from the analysis model of conventional TCP Westwood. The trade-off between congestion balance and protocol flappiness is parameterized. However, MPTCPW is not Pareto optimal because of design with such trade-off. The sender also uses two rules on path r as follows.

$$\text{Increase: } w_r \leftarrow w_r + \min(\delta_r/w_r, 1/w_r),$$
$$\text{Decrease: } w_r \leftarrow B_r \times RTT_{min,r},$$

where

$$\delta_r = w_r^{\gamma} \max_s \left\{ \frac{w_s^{2-\gamma}}{RTT_s^2} \right\} \Big/ \left(\sum\nolimits_i x_i \right)^2,$$

γ is a trade-off parameter between flappiness and congestion balance, $\gamma \in [0, 2]$. MPTCPW uses $\gamma = 1$.

To demonstrate lack of the congestion balance capability of wMPTCPW compared with MPTCPW, we used a simulation scenario as shown in Fig. 3(a), where link capacity and propagation delay of links 1 and 2 are set (40Mbps, 30ms) and (32Mbps, 90ms), respectively. To generate heavy load in link 1 and light load in link 2, we use single-path TCP Westwood (TCPW) [10] flows as the background traffic with twenty flows ($n = 20$) on link 1, and twelve flows ($m = 12$) on link 2. The observed packet loss rates at links 1 and 2 after experiment are 0.29% and 0.1%, respectively. Fig. 2(a) shows that the data rate of sub-flow on the congested and short RTT path (path 1) is very low. While the longer RTT path (path 2) has light load, the sub-flow on that path surges its rate up. Therefore, the total throughput of two sub-flows reaches the expected value as shown in Fig. 2(b), which is equivalent to the average rate of twelve TCPW flows that over the best path, path 2.

Fig. 2(c) shows that each sub-flow gets approximately one-half of TCPW flows' average rate that they are sharing. The packet loss rates at links 1 and 2 are measured to be 0.31% and 0.08%, respectively. Although path 1 is congested, the sub-flow on that path still sends blindly packets at haft rate. Therefore, wMPTCPW cannot perform load-balancing function compared to MPTCPW. Other comparisons MPTCPW with MPTCP-LIA are described details in [3].

III. Design of Multipath TCP Veno

A. Single-path TCP Veno Background

TCP-Veno [5] is combination of TCP Vegas [11] and TCP Reno [12] to take advantage of distinguishing packet losses caused by random error of wireless links or by network congestion. Its congestion window update rules are described as in Algorithm 1. $n\_backlog$ denotes the number of data packet backlogged at router's queue by estimating *Expected* and *Actual* rates as follows.

$$Expected = w/base\_rtt,$$
$$Actual = w/rtt,$$
$$n\_backlog = (Expected - Actual) \times base\_rtt,$$

where $base\_rtt$ is the minimum of RTTs during TCP connection.

Algorithm 1 Single-path TCP Veno algorithm [5].

Increase:
if $n\_backlog < \beta$ **then**
  /* Whenever receiving a new ACK */
  $w \leftarrow w + 1/w$
else
  /* Whenever receiving two new ACKs */
  $w \leftarrow w + 1/w$
end if

Decrease:
if $n\_backlog < \beta$ **then**
  /* Packet loss was caused by random error */
  $w \leftarrow w - w/5$
else
  /* Packet loss was caused by network congestion */
  $w \leftarrow w - w/2$
end if

B. Design of Multipath TCP Veno (MPVeno)

We take advantage of MPTCP-OLIA design technique in order to apply into Increase rule of MPVeno. A Sub-flow on path r adjusts its congestion window size according to Algorithm 2.

Algorithm 2 Multipath TCP Veno algorithm for the sender on path r.

Increase:
if $n\_backlog_r < \max(1, \theta_r\beta)$ **then**
  /* Whenever receiving a new ACK */
  $w_r \leftarrow w_r + a_r/w_r + \alpha_r/w_r$
else
  /* Whenever receiving two new ACKs */
  $w_r \leftarrow w_r + a_r/w_r + \alpha_r/w_r$
end if

Decrease:
if $n\_backlog_r < \max(1, \theta_r\beta)$ **then**
  /* Packet loss was caused by random error */
  $w_r \leftarrow w_r - w_r/5$
else
  /* Packet loss was caused by network congestion */
  $w_r \leftarrow w_r - w_r/2$
end if

In Algorithm 2, a_r and α_r are calculated as in Eqs. (2) and (3), respectively. To determine θ_r, we suppose that a N-path MPVeno flow were sharing a single bottleneck link with a single-path TCP Veno flow. Goal (i) implies that the total throughput of N-path MPVeno should be equivalent to that of single-path flow in such case. This also means that the total backlog of N sub-flows equals to that of single-path one (i.e., β packets). Meanwhile, each sub-flow r maintains $\theta_r\beta$ packets

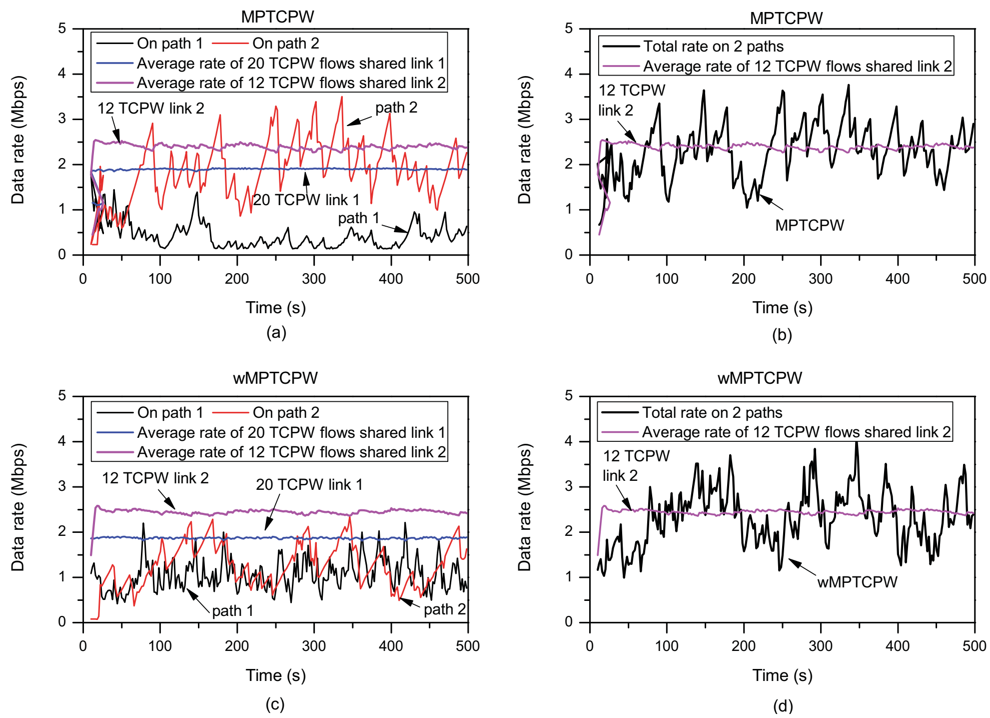

Fig. 2. (a) and (b) the congestion balance capability of MPTCPW; (c) and (d) Lack of the congestion balance capability of wMPTCPW.

at router' shared queue, therefore

$$\theta_r = x_r / \sum_i x_i.$$

This result conforms to design of multipath weighted Vegas as described in [13]. $\max(1, \theta_r\beta)$ guarantees that the backlog threshold always is at least one packet even it is possible θ_r very small.

IV. Performance Evaluations

In this section, we evaluate fair sharing of MPVeno with single-path TCP Veno, and compare its performance with MPTCP-OLIA [2] and MPTCPW [3] in wireless networks. In addition, we investigate friendliness of MPVeno with MPTCP-OLIA in wired networks as well. Experiments were run by NS-2 [14] with SACK option, active RED queue management, router buffer size of bandwidth-delay product, and data packet size of 1000 bytes.

A. Fair Sharing at a Single Bottleneck Link

In this section, we demonstrate how to share bandwidth fairly between a two-path MPVeno flow and a single-path TCP Veno flow at a single bottleneck link. The simulations used a dumbbell scenario as shown in Fig. 3(b). Fig. 4(a) shows the number of sent data of two protocols as a function of time. Because Increase rule of each MPVeno sub-flow is less aggressive than that of single-path TCP flow, the amount of data sent by the single-path TCP Veno flow approximates total of data sent by the two-path MPVeno flow as shown in Fig. 4 (b).

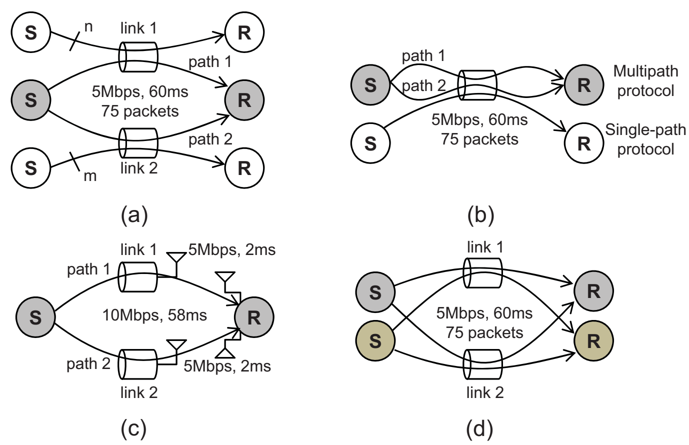

Fig. 3. Simulation scenarios.

B. Performance over Wireless Links

In this section, we investigate the performance of MPVeno compared with that of MPTCP-OLIA. The experiments were run on a two-separate path scenario as shown in Fig. 3(c) and wireless links' random error varies from 0.001% to 5%.

Fig. 5(a) shows that the average goodput of MPVeno is higher than that of MPTCP-OLIA when two paths are experienced the same packet loss rates from $5 \times 10^{-3}\%$ to 5%. Distinguishing the cause of the packet loss helps MPVeno to avoid unreasonable half reduction of congestion window over wireless links. Moreover, the performance of MPVeno depends on choice of threshold β shown as in Fig. 5. In Fig. 5 (a), the

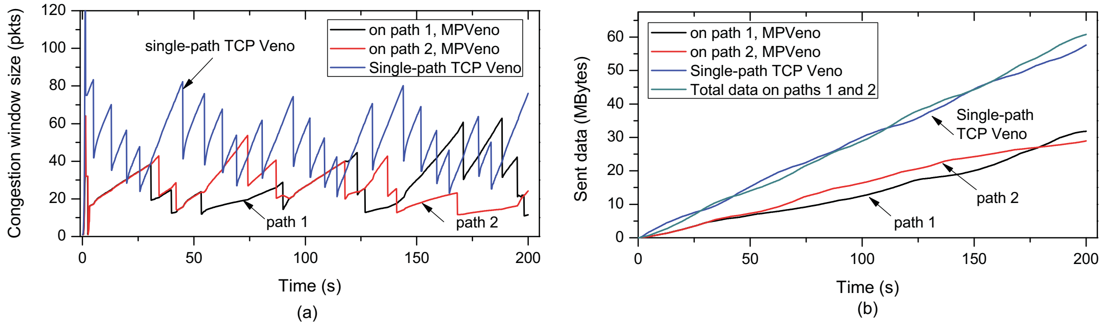

Fig. 4. (a) Evolution of congestion windows of a two-path MPVeno flow and a single-path TCP Veno flow, which are sharing at a single bottleneck link; (b) their amount of data transferred as function of time. $\beta = 6$ for MPVeno and TCP Veno.

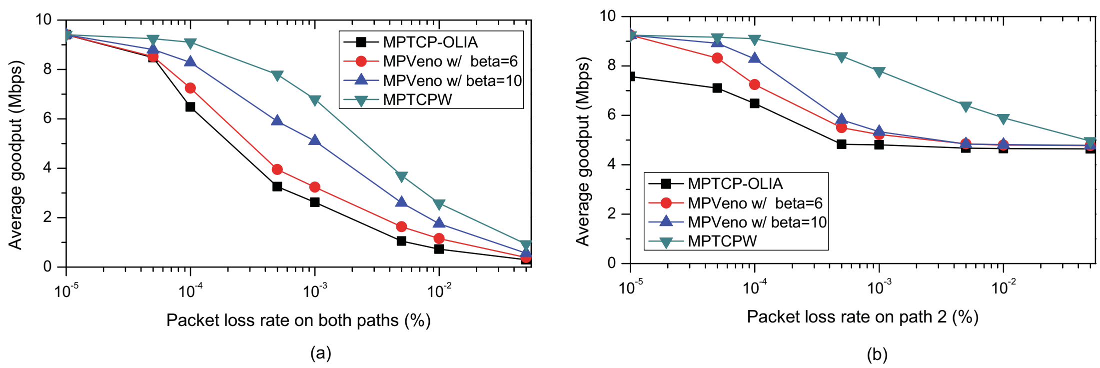

Fig. 5. (a) The average goodput of multipath protocols vs. varying packet loss on paths 1 and 2; (b) fixing random loss rate on path 1 at 0.01% while varying random loss rate on path 2.

goodput of MPVeno is improved more significantly as setting β to larger value. The larger β the higher network congestion threshold, this implies that working region for congestion window decrease by $w_r/5$ is larger, instead of decrease by $w_r/2$ as network congested.

In the next evaluation, we investigate the performance of MPVeno and MPTCP-OLIA when fixing the random loss rate for path 1 at 0.01% and varying the loss rate for path 2 from 0.001% to 5%. Fig. 5(b) shows that the goodput of MPTCP-OLIA is degraded dramatically when the random loss rate on path 2 increases to 1%. Since MPVeno can distinguish random loss from congestive loss, it outperforms at any condition. In Fig. 5(b), when random loss on path 2 is very high (greater than 1%), the total goodput of two-path transport protocols is dominated by only the best path, path 1.

In Fig. 5, the goodput of MPTCPW is highest among considered multipath protocols. However, we demonstrate in that MPTCPW cannot co-exist with MPTCP-OLIA in wired networks in next section.

C. Friendliness to existing MPTCP-OLIA flows

In this section, we examine friendliness of MPVeno and MPTCPW to existing MPTCP-OLIA in wired networks. A good multipath transport protocol is to fairly share bandwidth with MPTCP-OLIA in wired networks, and outperforms MPTCP-OLIA over wireless links. The simulations were run with scenario as shown in Fig. 3(d).

Fig. 6(a) shows that a two-path MPTCPW flow aggressively obtains more bandwidth than a two-path MPTCP-OLIA flow. The explanation for this result is that MPTCPW updates congestion window to estimated bandwidth instead of blindly halving congestion window as in MPTCP-OLIA. In contrast to MPTCPW, MPVeno is nice to share bandwidth with MPTCP-OLIA as shown in Fig. 6(b).

The evaluation of congestion balance of MPVeno gives similar results as that of MPTCP-OLIA, hence we do not report in this paper.

V. Conclusions

In this paper, we analyze and evaluate multipath congestion control algorithms (such as MPTCP-LIA, MPTCP-OLIA, MPTCPW, and wMPTCPW) over wireless networks, and propose MPVeno (an multipath transport version of TCP Veno) to improve multipath TCP performance for wireless environments. The simulation results show that MPVeno outperforms MPTCP-OLIA over wireless networks while preserving nice bandwidth sharing with conventional TCP Veno and MTCP-OLIA flows in wired networks.

Acknowledgments

This research was funded by Vietnams National Foundation for Science and Technology Development (NAFOSTED)

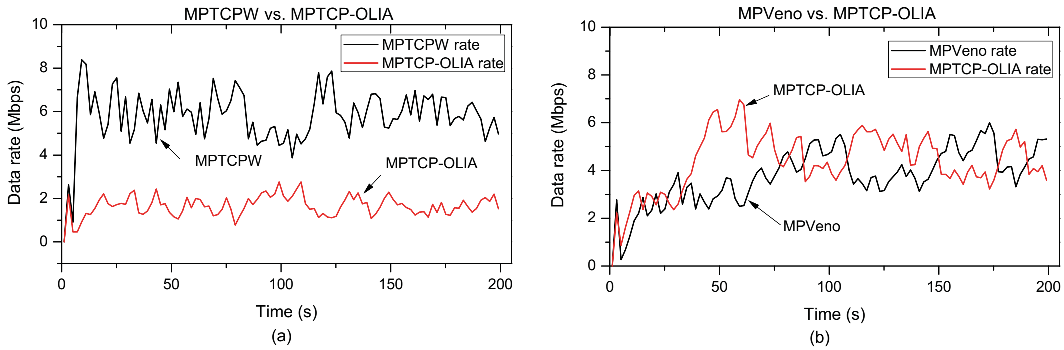

Fig. 6. (a) MPTCPW unfairly shares bandwidth with MPTCP-OLIA; (b) MPVeno fairly shares bandwidth with MPTCP-OLIA. Router buffer sizes are set to 75 packets (equals bandwidth-delay product).

under grant number 102.02-2013.48.

References

[1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," IETF RFC 6824, Jan. 2013.

[2] C. Raiciu, M. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," IETF RFC 6356, Oct. 2011.

[3] T. A. Le, C. seon Hong, and E.-N. Huh, "Coordinated TCP Westwood congestion control for multiple paths over wireless networks," in *Information Networking (ICOIN), 2012 International Conference on*, 2012, pp. 92–96.

[4] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec, "MPTCP is not pareto-optimal: performance issues and a possible solution," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 1–12.

[5] C. P. Fu and S. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 2, pp. 216–228, 2003.

[6] C.-M. Huang and C.-H. Tsai, "WiMP-SCTP: Multi-path transmission using stream control transmission protocol (SCTP) in wireless networks," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 1, 2007, pp. 209–214.

[7] C. Casetti and W. Gaiotto, "Westwood SCTP: load balancing over multipaths using bandwidth-aware source scheduling," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 4, 2004, pp. 3025–3029 Vol. 4.

[8] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. of the USENIX NSDI'11 Conf.*, Mar. 2011.

[9] H. Nuha, H. Hendrawan, and F. Yulianto, "Wireless multi-path TCP Westwood+ modification to achieve fairness in HSDPA," in *Computer Modeling and Simulation (EMS), 2010 Fourth UKSim European Symposium on*, 2010, pp. 402–407.

[10] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli, "Performance evaluation of Westwood+ TCP congestion control," in *Performance Evaluation*, Jan. 2004.

[11] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet," *Selected Areas in Communications, IEEE Journal on*, vol. 13, no. 8, pp. 1465–1480, 1995.

[12] V. Jacobson and M. J. Karels, "Congestion avoidance and control," in *ACM SIGCOMM*, 1988.

[13] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in *Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP)*, ser. ICNP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–10.

[14] NS-2 network simulator. [Online]. Available: http://www.isi.edu/nsnam/ns/