Multipath FAST TCP for Large Bandwidth-Delay Product Networks

Bich-Phuong Ha, Bao-Yen Tran, Tuan-Anh Le, and Cong-Hung Tran
Faculty of Information Technology,
Posts and Telecommunications Institute of Technology
Hochiminh City, Vietnam
Email: haphuongkm07@gmail.com, ttbyen.hcm@vnpt.vn, {letuanh,conghung}@ptithcm.edu.vn

***Abstract*— In such a rapid growing computer age, there are more and more applications that require to transmit data over large bandwidth-delay product (BDP) networks. Besides, with the increment of producing smart portable devices equipped multiple interfaces, several schemes, e.g., MPTCP, wVegas, have been proposed to exploit multi-home at two end-to-end hosts. These algorithms improve the performance and resilience via transport protocol, yet they cope with under-utilization available bandwidth in large BDP networks. In this paper, we propose MPFAST, which is an extension of FAST TCP for multiple paths. Based on FAST TCP, which uses queuing delay as a measure of congestion, it has the ability to fully utilize the resources in large BDP networks. The results of our simulation show that MPFAST can outperform these protocols with regard to throughput and fairness to regular FAST TCP flows.**

I. Introduction

In such a rapid growing computer age, more and more applications demand to transmit data over large BDP networks. However, the congestion control (CC) algorithm of current TCP referred as TCP Reno [1] is proved to be less efficient in utilizing the network resources, especially when working in the networks mentioned above [2]. Thus, in order to improve the connection's throughput, HSTCP and Scalable TCP were proposed in [2],[3], respectively. These schemes, as pointed out in [4], are far more aggressive than TCP Reno which results in their much worse intra-protocol fairness and TCP fairness. Therefore, FAST TCP [4] was introduced to solve the main problems of TCP and enhance the throughput, fairness, steadiness and reactiveness compared to those algorithms in high-speed and/or long-latency networks. Using queuing delay as a congestion signal gives more accurate estimated information and has natural scaling with network capacity.

Meanwhile, the increase of producing smart portable devices with multiple interfaces urges many researchers to seek for the schemes that efficiently make use of network resources. Hence, approved by the IETF, MPTCP [5] improves the connections throughput by allowing a single path to be split into multiple sub-flows so that it utilizes the networks more effectively than regular TCP. Furthermore, it is recently considered promising for data center networks load balancing [6]. However, designed to be accordant with regular TCP, MPTCP also has to face the well-known problems while working in large BDP networks [7]. Lately, another proposal is weighted Vegas (wVegas) [8], which is a delay-based CC for multipath TCP. It has been proven to improve the fairness and efficiency of traffic shifting, even so, it is originated from TCP Vegas which leads to the same problems as MPTCP. That is because both are designed for small BDP paths so that they are less efficient in large BDP paths. By contrast to wVegas and MPTCP, MPCubic [9] has been recently proposed for large BDP networks, it shows an improvement in throughput, load balancing, and fairness. However, it belongs to loss-based approach using packet loss as a sign to decrease transmission rate, which achieves much less fine-grained of load balancing than delay-based approach.

In this paper, we propose another delay-based approach for multipath transfer protocol, called MPFAST. This paper studies its performance in terms of three design goals for a multi-path TCP CC, which are (1) efficiency, (2) fairness, and (3) congestion balance [5]. The simulation results show that MPFAST can outperform the aforementioned algorithms, simultaneously it is promising for effectively utilizing the resources in large bandwidth and/or long delay networks.

The rest of paper is organized as follows. In next section, we express concisely the related work. Then, we describe details of MPFAST design in III. Performance evaluation is discussed in IV and a brief conclusion in V.

II. Related Work

In order to shift traffic onto least congested paths while maintaining a good fairness, responsiveness and stability, there are various proposals to adding to multipath protocol. Howbeit, none of these is a fine-grained and large BDP design-based approach. We summarize these algorithms as follows.

pTCP [10], CMT over SCTP [11] and M/TCP [12] perform uncoupled congestion control on each path. These mechanisms is unfair to single-path when they competing.

After investigating the behaviors of MPTCP with linked increase algorithm (LIA) through different scenarios, R. Khalili and N. Gast et.al pointed out that it suffers from two following problems [14]. Firstly, there would be no benefit to upgrading TCP users when they upgrade to MPTCP even if this can cause descent in throughput of other users, which is called non-Pareto optimality symptom. Secondly, MPTCP is unreasonably aggressive towards TCP. The reason is that, LIA design forces a tradeoff between responsiveness and resource pooling. So as to provide good responsiveness, the current implementation

978-1-4799-3792-9/14/$31.00 ©2014 IEEE

of LIA must depart from Pareto-optimality making its miss achieving its design goals.

Most existing multipath proposals use packet loss that leads to a result of coarse-grained load balancing due to less estimated congestion information.

A recent research, weighted Vegas (wVegas) [8] has solved the problems of multipath CC by using queuing delay as a congestion signal, thus achieving fine-grained load balancing and better RTT fairness. However, wVegas is only appropriate to slow-speed and/or short-delay networks. According to [8], it is less efficient on large BDP paths.

MPCubic [9], a loss-based multipath TCP, has been developed from Cubic TCP, which is a transport protocol designed for large BDP networks. MPCubic is not only a fair sharing algorithm but also a fast recovery scheme. Nonetheless, it is not a delay-based approach which is considered as an improvement of loss-based approach of round-trip time (RTT) fairness.

III. Design of Multipath FAST TCP

A. Regular Single-Path FAST TCP

In this section, we briefly summarize the FAST TCP algorithm for single-path transmission, a TCP congestion control algorithm for high-speed and/or long-latency networks. Let us introduce the following equation of FAST TCP which adapts the congestion window size.

Consider a source i of single-path FAST TCP [4] adjusts its congestion window size $w_i(t)$ periodically as follows:

$$w_i(t+1) = \gamma\left(\frac{d_i w_i(t)}{d_i + q_i(t)} + \alpha_i\right) + (1-\gamma)w_i(t), \qquad (1)$$

where $\gamma \in (0,1]$, d_i and q_i denote the round-trip propagation delay and the queueing delay observed at source i, respectively. α_i (constant) denotes the number of packets of source i backlogged at routers along its route. Note that $T_i(t) := d_i + q_i(t)$ is RTT measured at source i, and $x_i(t) := w_i(t)/T_i(t)$ is data rate of source i at time t.

We rewrite (1) as

$$w_i(t+1) = w_i(t) + \gamma\left(\alpha_i - x_i(t)q_i(t)\right). \qquad (2)$$

For single-path FAST TCP, the equilibrium value of $x_i(t)$ derived from (2) is

$$\hat{x}_i = \frac{\alpha_i}{\hat{q}_i}, \qquad (3)$$

where $\hat{q}_i$ denotes the equilibrium value of $q_i(t)$.

B. Design of Multipath FAST TCP for Concurrent Transfer

We now propose a multipath version of single-path FAST TCP (called MPFAST) in this section. The design of MPFAST is started from the single-path model as follows. Let denote r a sub-flow of MPFAST which sends its data packets on path r. MPFAST's congestion control algorithm of source s adapts the congestion window size $w_{s,r}$ on path r periodically according to

$$w_{s,r}(t+1) = \gamma\left(\frac{d_{s,r} w_{s,r}(t)}{d_{s,r} + q_{s,r}(t)} + \theta_{s,r}(t)\alpha_s\right) + (1-\gamma)w_{s,r}(t), \qquad (4)$$

where $\theta_{s,r}(t) \in [0,1]$ (note that $\sum_r \theta_{s,r} = 1$) denotes the weight of source s on path r. Let us define N to be the number of sub-flows (i.e., paths) of a MPFAST connection. Similarly, the equilibrium data rate of sub-flow r of source s derived from (4) is

$$\hat{x}_{s,r} = \frac{\hat{\theta}_{s,r}\alpha_s}{\hat{q}_{s,r}}. \qquad (5)$$

The equilibrium total rate of source s $y_s(t)$ is defined by

$$\begin{aligned} \hat{y}_s &:= \sum_{r=1}^{N} \hat{x}_{s,r} \\ &= \alpha_s \sum_{r=1}^{N} \frac{\hat{\theta}_{s,r}}{\hat{q}_{s,r}}. \end{aligned} \qquad (6)$$

To determine $\hat{\theta}_{s,r}$, we consider all the paths experienced the same queuing delay, denoted by q_s. Then, (6) becomes

$$\begin{aligned} \hat{y}_s &= \alpha_s \sum_{r=1}^{N} \frac{\hat{\theta}_{s,r}}{\hat{q}_{s,r}} \\ &= \frac{\alpha_s}{\hat{q}_s} \sum_{r=1}^{N} \hat{\theta}_{s,r} \\ &= \frac{\alpha_s}{\hat{q}_s} \qquad \text{since} \sum_{r=1}^{N} \hat{\theta}_{s,r} = 1, \end{aligned} \qquad (7)$$

where $\hat{\theta}_{s,r}$, $\hat{q}_{s,r}$, and $\hat{y}_s$ denote the equilibrium values of $\theta_{s,r}(t)$, $q_{s,r}(t)$, and $y_s(t)$, respectively. And (5) becomes

$$\hat{x}_{s,r} = \frac{\hat{\theta}_{s,r}\alpha_s}{\hat{q}_s}. \qquad (8)$$

By substituting (7) into (8), we have

$$\hat{\theta}_{s,r} = \frac{\hat{x}_{s,r}}{\hat{y}_s}. \qquad (9)$$

The equations (5) and (9) can be interpreted in each sub-flow on path r which attempts to inject the number of packets in its paths so that its packet backlogged at routers' queue along path r are proportional to $\hat{\theta}_{s,r}\alpha_s$. For Goal 2, assume that a two-path MPFAST flow is competing with a single-path FAST TCP flow at a shared bottleneck link, two sub-flows and single-path flow will be experienced the same queuing delay. Because α_i (for single-path flow) and α_s (for multipath flow) are identical constants, the data rates of two sub-flows are split by factor $\hat{\theta}_{s,r}$. In such case, $\hat{\theta}_{s,r}$ approximates $1/2$, so total throughput of two-path flow will be equivalent to those single-path flow.

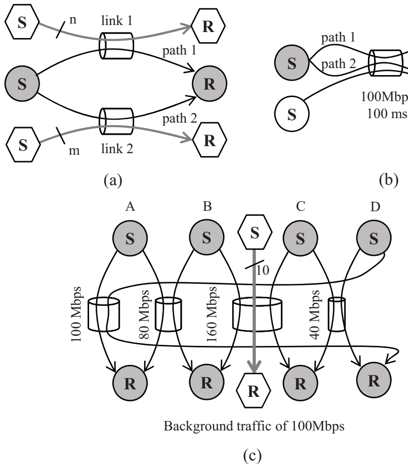
The sub-flow on path r changes its congestion window size described by the pseudo-code of MPFAST shown as in Algorithm 1.

IV. Performance Evaluations

In this section, we investigate the behaviors of MPFAST with aspects of efficiency, fairness, and congestion balance. Our simulation experiments use the topologies shown in Fig.1, are run on NS-2 [15] with SACK option, data packet size of 1448 bytes, and router buffer size of BDP. We take samples of data rate at every 2 seconds.

Algorithm 1 Multipath FAST TCP algorithm for the sender on path r.

```
Initialization: $total\_rate \leftarrow 0$
On receipt of each update of a subflow r:
$Wq[r] = W[r] * (avgRTT[r] - baseRTT[r]);$
if ($Wq[r]! = \max(1, \hat{\theta}[r] * \alpha) * avgRTT[r]$) then
    $targetW[r] = (1 - \gamma) * cwnd[r] +$
        $\gamma * (W[r] * (baseRTT[r]/avgRTT[r]) + \hat{\theta}[r] * \alpha);$
    $targetW[r] = max(2, targetW[r]);$
    $W[r] = cwnd[r];$
    /* call function fast_pace() in regular FAST TCP */
    $fast\_pace(targetW[r] - cwnd[r]);$
end if
/* Calculate smoothed rate for each sub-flow */
$rate[r] = 0.875 * rate[r] + 0.125 * cwnd[r]/rtt[r];$
$total\_rate+ = rate[r];$
/* Calculate $\hat{\theta}[r]$ for each sub-flow*/
$\hat{\theta}[r] = rate[r]/total\_rate;$
```



(a)

(b)

(c)

Fig. 1. Simulation topologies.

A. Efficiency

In this section, we demonstrate MPFASTs throughput improvement, compared to wVegas in a scenario shown in Fig.1(a), where two links have the same configuration with links capacity of 80 Mbps and propagation delay of 50ms; background traffic is generated by eight constant bit-rate (CBR) flows of 5 Mbps on each link (n = m = 6). They are started from 100s on path 1 and 120s on path 2, they are stopped at 200s. On each path, a CBR flow is started after another 3 seconds.

Fig. 2(a) depicts that both two wVegas sub-flows throughput increase slowly in the environment without background traffic, and it takes a long time to reach the peak rate (in this case is 80 Mbps). When background traffic is started at 100s on path 1 and 120s on the remaining, the rates of sub-flow 1 and 2 go down immediately and keep stability. After stopping background flows at 200s on both paths, two sub-flows obtain the expected throughput (i.e., 80 Mbps) toughly and need longer time than MPFAST which will be mentioned following.

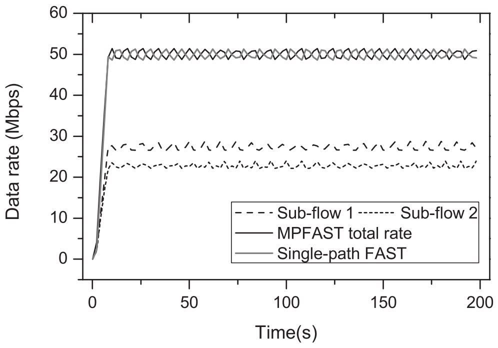


Fig. 3. MPFAST performs good fairness to a single-path FAST flow over single bottleneck link.

In contrast, as shown in Fig.2(b) two MPFAST sub-flows need a very short time to achieve the expected throughput. When the background traffic appears on both paths at the same time we mentioned above, both sub-flows' throughput decline slowly and remain stably. Its throughput maintains at a same level which is about a half of perfect rate. When we stop background traffic at 200s on both paths, throughput drastically get its last maximum rate.

B. Fairness to single-path FAST TCP

We now investigate MPFASTs fairness to regular FAST TCP at common bottleneck link. In this experiment, a two-path MPFAST flow shares with a regular FAST TCP flow at a link as shown in Fig.1(b), with links capacity of 100 Mbps and propagation delay of 100 ms. Fig.3 compares the throughput of two sub-flows and one regular FAST TCP flow. As an overall trend, the regular FASTs window size is almost twice of the sub-flow. Therefore, the average throughput of two-path MPFAST flow is equivalent to that of FAST TCP flow. Although in the same network congestion, sub-flow 1 and 2 do not share bandwidth equally because their average $\hat{\theta}$s are measured about 0.54 and 0.45, respectively. There is a very small difference between $\hat{\theta}$ parameters in the experiment and the value in theory which is 0.5. Though, we believe this result is reasonable as it is difficult to obtain the perfect sharing in reality of the network. In brief, the above results imply that MPFAST can fairly share with regular FAST TCP at the single bottleneck link so that it satisfies the second goal design of a multipath TCP congestion control.

C. Congestion Balance

To investigate the effectiveness of resource pooling and traffic shifting of MPFAST, we use the topology in Fig.1(c) with propagation delay is 50 ms. We then start ten CBR flows with 10 Mbps at 100s in order to reduce bandwidth available and turn them off at 200s.

Fig.4 shows the good ability of resource pooling and traffic

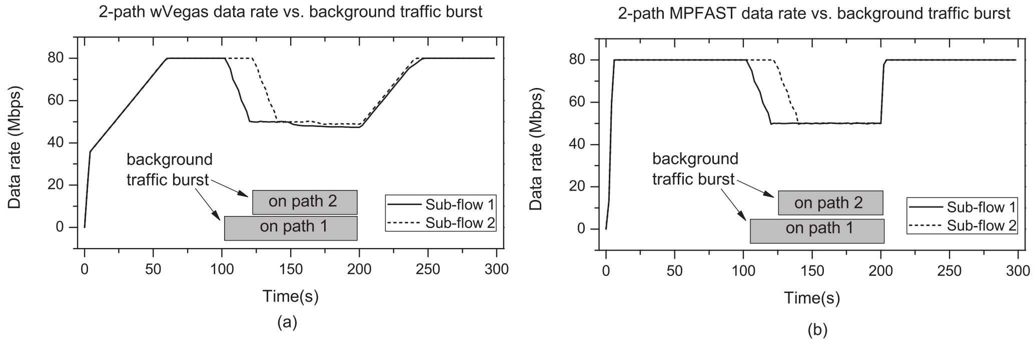


Fig. 2. Effectiveness of multipath protocols for large BDP network when background traffic appears. (a) two-path wVegas responses slowly; (b) two-path MPFAST responses rapidly.

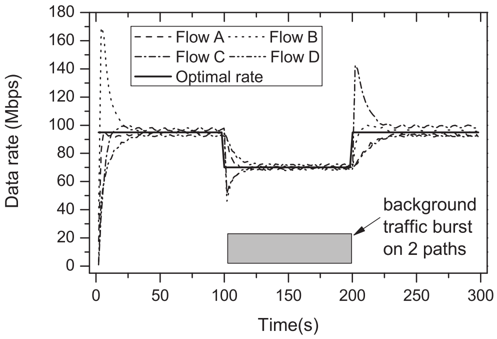


Fig. 4. Pooling network bandwidth resources and distributing equally to four multipath flows according to network changes.

shifting onto less congested paths of MPFAST. As expected, each flow firstly tends to increase quickly as much as they can, then maintain around the optimal rate of 95 Mbps[1]. By the time all the background traffic flows started running, there is a considerable drop in their data rate, and four flows remain steady at the rate of 70 Mbps after 8s. Their data rate go up again but less steeply at which point the background traffic are turned off, and stay constant nearly at the first optimal rate.

In summary, MPFAST shows how well it can balance the congestion in the network by shifting traffic away from more congested paths and keep the stability in its sending rate.

V. Conclusions

We propose a multipath congestion control algorithm that can split its traffics across multiple available paths into multiple sub-flows for improving efficiency, fairness and load balancing for large BDP networks, called MPFAST. It is based on FAST which uses queuing delay as congestion signal, therefore, it could efficiently utilize the network resources. Through simulation results, we found that MPFAST has remarkable contribution on transmission between end-to-end hosts.

Acknowledgments

This research was funded by Vietnams National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2013.48.

References

[1] V. Jacobson and M. Karrels, "Congestion avoidance and control," *SIGCOMM Comput. Commun. Rev.*, 1988.

[2] S. Floyd, "HighSpeed TCP for large congestion windows," *SIGCOMM Comput. Commun. Rev.*, Dec 2003.

[3] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *in First International Workshop on Protocols for East Long Distance Networks*, Feb 2003.

[4] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, architeccture, algorithm, performance," in *IEEE Infocom*, 2004.

[5] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath tcp development," IETF RFC 6824, 2011.

[6] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," 2011.

[7] D. Wischik, C. Raiciu, and M. Handley, "Balancing resource pooling and equipoise in multipath transport," *University College London*, 2010.

[8] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath tcp," *IEEE ICNP*, 2012.

[9] T. Le, R. Haw, C. Hong, and S. Lee, "A multipath Cubic TCP congestion control with multipath fast recovery over high bandwidth-delay networks," *IEICE Trans. Comm*, pp. 2232–2244, 2012.

[10] H. hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," 2002, pp. 83–94.

[11] J. Iyengar, P. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent and-to-end paths," *IEEE /AMC Trans. Netw*, 2006.

[12] K. Rojviboonchai and H. Aida, "An evaluation of multi-path transmission control protocol m/tcp with robust acknowledgment schemes," *IEICE Trans. Comm*, 2004.

[13] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," 2004.

[14] 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.

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

[1] With the topology in Fig.1(c), four MPFAST flows will add up the bandwidth of all links as a single virtual link of 380 Mbps. Therefore, the optimal rate for each flow is calculated by dividing 380 Mbps by 4 flows.