MPCubic: An Extended Cubic TCP for Multiple Paths over High Bandwidth-Delay Networks

Tuan Anh Le, Choong Seon Hong, and Sungwon Lee

Department of Computer Engineering, Kyung Hee University
1 Seocheon, Giheung, Yongin, Gyeonggi, 446-701, Korea
{letuanh, cshong, drsungwon}@khu.ac.kr,

***Abstract*— Cubic TCP nowadays is one of transport protocols designed for high bandwidth-delay networks has successfully deployed in the Internet. Multi-homed computers with multiple interfaces to access the Internet via high speed links will become more popular. In this work, we introduce an extended version of Cubic TCP for multiple paths, called MPCubic. The extension process is approached from analysis model of Cubic with constraining coordinated congestion control between paths, and fairness to regular Cubic. Therefore MPCubic can spread its traffic across paths in load-balancing manner, while preserving fair sharing with regular TCP, Cubic, and MPTCP at common bottlenecks. Our simulation results show that MPCubic can achieve stability, throughput improvement, fairness, and load-balancing.**

I. INTRODUCTION

Cubic TCP[1] [2] nowadays is one of transport protocols over high BDP networks has successfully deployed in the Internet. Its congestion window growth is implemented by a cubic function, which increases congestion window in terms of concave and convex curves. Such congestion window growth depends on the time between two consecutive congestion events. Hence Cubic not only improves performance in high BDP networks but also fairly shares at common bottleneck with heterogeneous round trip time (RTT) Cubic flows.

In addition, multi-homed computers with multiple interfaces to access the Internet via high speed links will become more popular. This would promise a big chance in order to improve the performance and resilience via transport protocol if multiple links was used simultaneously [3].

Multipath TCP protocol allows to create simultaneous multiple sub-flows amongst two end hosts through multiple network interfaces [4] where each sub-flow maintains sending data packets over a path. Coordinated multipath-aware congestion control (MPTCP) [5], which is in the standardization process by IETF, sees a set of resources on all paths as a single resource as suggested in the resource pooling principle [6]. The coordinated congestion control moves more traffic on the less-congested paths as load balancing mechanism. MPTCP was designed to be backward-compatible with regular TCP (TCP Reno) [7], which has been proven low utilization of bandwidth available in high bandwidth-delay product (BDP) networks. So far, there is no transport protocol supporting multiple paths over high BDP networks. Therefore an urgent request is to design/extend a multipath transport protocol for high BDP networks.

In this work, we introduce an extended version of Cubic TCP [2] for multiple paths, called MPCubic. The extension process is approached from the analysis model of Cubic. By establishing coordinated congestion control between paths and constraining fairness to regular single path Cubic per-path and per-flow, our MPCubic is able to move traffic away from congested paths to uncongested paths, and fairly share to regular Cubic and TCP at common bottleneck. Our simulations show that MPCubic can achieve stability, throughput improvement, fairness, and load-balancing.

The rest of this paper is organized as follows. Section II gives related work which summarize previous works relevant to multipath congestion control. We present the details of the extension process to MPCubic, and MPCubic algorithm implemented by pseudocode in Section III. Section IV presents the results of simulation evaluation in terms of window evolution fluctuation, fairness, improved throughput and resource pooling. Finally, Section V gives some conclusions of our current work and future work.

II. RELATED WORK

In the section, we summarize several solutions for multipath congestion control problem.

Multiple paths TCP (mTCP) [8] focuses on striping data packets across multiple paths, and detecting common bottleneck to alleviate unfair sharing by computing correlation between fast retransmit intervals.

A sub-flow in Parallel TCP (pTCP) [9], concurrent transfer multipath (CMT) over SCTP [10] sends data packets independently each others because it uses uncoordinated congestion control algorithm (using regular TCP). These protocols can not fairly share with regular TCP flows at common bottleneck since they don't handle/detect the common bottleneck.

Reliable Multiplexing Transport Protocol (R-MTP) [11] was designed for wireless networks. R-MTP's sub-flow tracks packet arrival time to estimate the available bandwidth for rate control, and for congestion detection.

III. EXTENSION PROCESS AND ALGORITHM

MPTCP was designed to ensure three goals [5] [12]: improve throughput, fairness, and load-balancing (resource pooling). MPTCP' coordinated congestion control is derived

---

[1] The previous version of Cubic, called BIC-TCP, [1] have been the default TCP algorithm in Linux since 2006.

from the fluid model in [13, equation (3)], where coordinated control between paths are implemented in the additive increase function with a constant increase factor. Since Cubic grows its window up based on the time between two consecutive congestion events [2], we may not apply the technique above for extending regular Cubic. We agree that any multipath transport protocol should satisfy those goals. Hence our extension process should not break down those.

Now, we consider sub-flow's congestion window growth on path s as follows

$$w_s(t) = \min\left(\delta C(t_s - K_s)^3 + W_s^{max}, C(t_s - K_s)^3 + W_s^{max}\right) \tag{1}$$

where C denotes a constant in regular Cubic, δ is a linking parameter between paths, t_s denotes the time elapsed from the last packet loss event on path s, and K_s denotes a period of time between two consecutive packet loss events (called a *congestion epoch*). To ensure fairness, MPCubic's window increment per path did not exceed that of regular Cubic by using the right argument in $\min(.)$.

We assume that a packet loss event is occurred on path s as the congestion window reaches W_s^{max}. It is then decreased by factor β as used in regular Cubic [2]. Therefore, we calculate K_s such that

$$\begin{aligned}
&\min\left(-\delta C K_s + W_s^{max}, -C{K_s}^3 + W_s^{max}\right) = (1-\beta)W_s^{max} \\
&\Rightarrow K_s = \left(\frac{\beta W_s^{max}}{\min(\delta,1)C}\right)^{1/3}.
\end{aligned} \tag{2}$$

To determine δ, we should analyze MPCubic performance model in the steady state. We consider MPCubic operating in the congestion window increase phase of the concavity mode as in [2]. The number of round trip times ($RTTs$) in a congestion epoch on path s is K_s/RTT_s, then the number of packets sent in that congestion epoch is

$$\begin{aligned}
&\min\left(\sum_{i=0}^{\frac{K_s}{T_s}}[\delta C(iT_s - K_s)^3 + W_s], \sum_{i=0}^{\frac{K_s}{T_s}}[C(iT_s - K_s)^3 + W_s]\right) \\
&= \sum_{i=0}^{\frac{K_s}{T_s}}[\min(\delta^3,1)C(iT_s - K_s)^3 + W_s] \\
&= \frac{W_s K_s}{T_s} - \min(\delta,1)C\sum_{i=0}^{\frac{K_s}{T_s}}(K_s - iT_s)^3 \\
&= \frac{W_s K_s}{T_s} - \min(\delta,1)C\sum_{i=0}^{\frac{K_s}{T_s}}(iT_s)^3 \\
&= \frac{W_s K_s}{T_s} - \min(\delta,1)CT_s^3\frac{\left(\frac{K_s}{T_s}(1+\frac{K_s}{T_s})\right)^2}{4} \\
&\approx \frac{W_s K_s}{T_s} - \min(\delta,1)CT_s^3\frac{\left(\frac{K_s}{T_s}\right)^4}{4} \\
&= \frac{W_s K_s}{T_s} - \min(\delta,1)\frac{CK_s^4}{4T_s},
\end{aligned} \tag{3}$$

where iT_s denotes the i^{th} RTT_s of a sub-flow in a congestion epoch on path s, W_s denotes the last congestion window size before a packet loss event. When the congestion window reaches W_s at the end of the congestion epoch, one of these packets on path s will be dropped with the packet loss probability p_s. Therefore, (3) must be

$$\frac{W_s K_s}{T_s} - \min(\delta,1)\frac{CK_s^4}{4T_s} = \frac{1}{p_s}. \tag{4}$$

By substituting (2) into (4), we obtain W_s as follows

$$W_s = \left(\min(\delta,1)\frac{C}{\beta}\right)^{1/4}\left(\frac{4T_s}{(4-\beta)p_s}\right)^{3/4}. \tag{5}$$

For the deterministic model, the average congestion window $\overline{w}_s$ on path s is calculated according to the number of packets sent over the number of RTTs for one congestion epoch as follows

$$\begin{aligned}
\overline{w}_s &= \left(\frac{W_s K_s}{T_s} - \min(\delta,1)\frac{CK_s^4}{4T_s}\right)/(K_s/T_s) \\
&= \left(\frac{4-\beta}{4}\right)W_s \\
&= \left(\min(\delta,1)\frac{C(4-\beta)T_s^3}{4\beta p_s^3}\right)^{1/4}.
\end{aligned} \tag{6}$$

If a Cubic flow ran on path s, its average window $\overline{w}_s^{Cubic}$ [2] would be

$$\overline{w}_s^{Cubic} = \left(\frac{C(4-\beta)T_s^3}{4\beta p_s^3}\right)^{1/4}. \tag{7}$$

Equations (6) and (7) give the relationship between MPCubic and Cubic as flows

$$\overline{w}_s = \overline{w}_s^{Cubic}\min(\delta^{1/4},1),$$

or

$$\overline{w}_s^{Cubic} = \max\left(\frac{\overline{w}_s}{\delta^{1/4}}, \overline{w}_s\right). \tag{8}$$

The improve throughput and fairness goals suggest that the total throughput of a multipath Cubic flow should equal that of a Cubic flow on the best path for it [13], i.e.,

$$\sum\nolimits_s \frac{\overline{w}_s}{RTT_s} = \max\nolimits_s\left\{\frac{\overline{w}_s^{Cubic}}{RTT_s}\right\}. \tag{9}$$

By substituting (8) into (9), we obtain

$$\sum\nolimits_s \frac{\overline{w}_s}{RTT_s} = \max\nolimits_s\left\{\max\left(\frac{\overline{w}_s}{\delta^{1/4}RTT_s}, \frac{\overline{w}_s}{RTT_s}\right)\right\}.$$

Since $\overline{w}_s$ never equals zero, equality of above equation cannot occur, therefore

$$\sum\nolimits_s \frac{\overline{w}_s}{RTT_s} = \max\nolimits_s\left\{\frac{\overline{w}_s}{\delta^{1/4}RTT_s}\right\}.$$

By solving for δ, we obtain

$$\delta = \frac{\max_s\left\{\left(\frac{\overline{w}_s}{RTT_s}\right)^4\right\}}{\left(\sum_s \frac{\overline{w}_s}{RTT_s}\right)^4}.$$

The literature [12] suggests that if δ was based on paths with large throughput (i.e., $\overline{w}_s/RTT_s$) rather than paths with large congestion window size $\overline{w}_s$, MPCubic could not achieve load-balancing since the lightly load paths give larger window sizes. Therefore, δ must be based on both the congestion window size and throughput. So we slightly modify δ, and then replace δ with δ_s,

$$\delta_s = \overline{w}_s^k \frac{\max_s\left\{\frac{\overline{w}_s^{4-k}}{RTT_s^4}\right\}}{\left(\sum_s \frac{\overline{w}_s}{RTT_s}\right)^4}.$$

where k is a trade-off parameter between resource pooling and protocol fluctuation, $0 \leq k \leq 4$. The fluctuation phenomena occurs for any multipath transport protocol when the multipath protocol sends packets on one path for moment, then it switches sending on the other, and so on [13] [12]. So resource pooling of any multipath transport protocol is adversely affected in such condition [12].

The following experiment results present the choice of k such that achieving good resource pooling but less fluctuation. In this paper, some results are obtained by comparing MPCubic with a uncoordinated multipath Cubic (called unMPCubic), where the congestion control algorithm (using regular Cubic) on each path operates independently each other. For two-path unMPCubic flow, we set the window increase scale factor of $(1/2)^{4/3}$ such that each sub-flow gets half the throughput of what a regular Cubic gets. Experiments were run on the network configuration with propagation delay of 100ms as shown in Fig. 2(a). The exogenous drops are regarded as the uniform random process. Fig. 1 shows the congestion window increase stages on paths 1 and 2 ($cwnd_1$, $cwnd_2$). The top plots in Fig. 1 show that when $p_1 = p_2$, unMPCubic demonstrates non-variation in increment of both windows; MPCubic with $k \leq 2$ gives the slight variations, and for $k = 3$, pairs of window increment move from one axis to another, and so on. Whereas when $p_1 > p_2$ as shown in the bottom plots in Fig. 1, MPCubic demonstrates the capability of less aggressive increase in the congestion window on path 1 and the more aggressive increase on path 2 as increasing k value. There is a reasonable trade-off between fluctuation and load-balancing when we choose $k = 2$.

For MPCubic operating in TCP-friendly region, we must calculate δ_s^{Tcp} separately in such that MPCubic still exists with not only Cubic flows but also TCP (TCP Reno) flows. The average congestion window $\overline{w}_s^{MTcp}$ of TCP flow [7] on path s is

$$\overline{w}_s^{Tcp} = \left(\frac{\alpha(2-\beta)}{2\beta p_s}\right)^{1/2}, \tag{10}$$

where α denotes the increase factor for each RTT, β denotes the multiplicative factor for a packet loss event. The congestion window of multipath TCP on path s increases by $\min\left(\delta^{MTcp}\alpha, 1\right)$ for each RTT, and decreases as the single-path TCP does. Hence, $\overline{w}_s^{MTcp}$ in the steady state is

$$\overline{w}_s^{MTcp} = \left(\min(\delta^{MTcp}, 1)\frac{\alpha(2-\beta)}{2\beta p_s}\right)^{1/2}. \tag{11}$$

By substituting (10) into (11) and then solving for δ^{MTcp} with the constrain similar to (9) but for $\overline{w}_s^{Tcp}$, we obtain

$$\delta^{MTcp} = \max_s\left\{\left(\frac{\overline{w}_s}{RTT_s}\right)^2\right\} / \left(\sum\nolimits_s \frac{\overline{w}_s}{RTT_s}\right)^2.$$

Although our extension approach is different from MPTCP design[2] [5], we should choose δ^{MTcp} in TCP-friendly region such that the throughput of both approaches in the steady state are the same. Therefore, we adopt δ_s^{MTcp} as

$$\delta_s^{MTcp} = \overline{w}_s \frac{\max_s\left\{\frac{\overline{w}_s}{RTT_s^2}\right\}}{\left(\sum_s \frac{\overline{w}_s}{RTT_s}\right)^2}.$$

Regular TCP (with $\alpha = 1$, $\beta = 1/2$) in the steady state gives $\overline{w}_s^{Tcp} = (3/2p_s)^{1/2}$. Therefore, multipath TCP's increase factor will be equivalent to that of regular TCP if it is

$$\left(\frac{\alpha(2-\beta)}{2\beta p_s}\right)^{1/2} = \left(\min(\delta_s^{MTcp}, 1)\frac{3}{2p_s}\right)^{1/2}$$
$$\Rightarrow \alpha = \min(\delta_s^{MTcp}, 1)\frac{3\beta}{2-\beta}.$$

Finally, we present multipath TCP's window growth function in MPCubic with respect to the elapsed time t on path s as

$$w_s^{MTcp}(t) = (1-\beta)W_{max} + \min(\delta_s^{MTcp}, 1)\frac{3\beta}{2-\beta}\frac{t}{RTT_s}. \tag{12}$$

IV. PERFORMANCE EVALUATION

In this section, we evaluate stability, throughput, fairness and load-balancing of MPCubic. Simulation experiments were run on NS-2 [14]. In all simulations, the transport protocols is set by the selective acknowledgment (SACK) option [15] and a 1000-byte data packet. We use the droptail discipline to demonstrate the evolution of congestion window as shown in Fig. 2(a). The other experiments use the random early detection (RED) queue management [16] at bottleneck routers to prevent them from synchronized packet drops. Data rates were sampled every 5 seconds.

A. *MPCubic congestion window evolution*

In this section, we demonstrate MPCubic congestion window evolution compared with that of a regular single-path Cubic flow. In our experiment, a two-path MPCubic flow was run on separate paths with the same network parameters as shown in Fig. 2(a). A regular single-path Cubic flow was separately simulated only one path. Fig. 3 shows that the regular Cubic flow is more aggressive in window growth than two-path MPCubic flow. We define a window growth cycle including two consecutive congestion epoches: a concave region and a concave-convex region. Fig. 3 shows that a

---

[2] The increase function of MPTCP on path s increases $\min\left(\delta/sumw(t), 1/w_s(t)\right)$ for each received positive ACK, with $\delta = \overline{sumw}\max_s\{\overline{w}_s/RTT_s^2\}/\left(\sum_s \overline{w}_s/RTT_s\right)^2$, where $sumw(t) = \sum_s w_s(t)$, $\overline{w}_s$ and $\overline{sumw}$ are average value of $w_s(t)$ and $sumw(t)$, respectively.

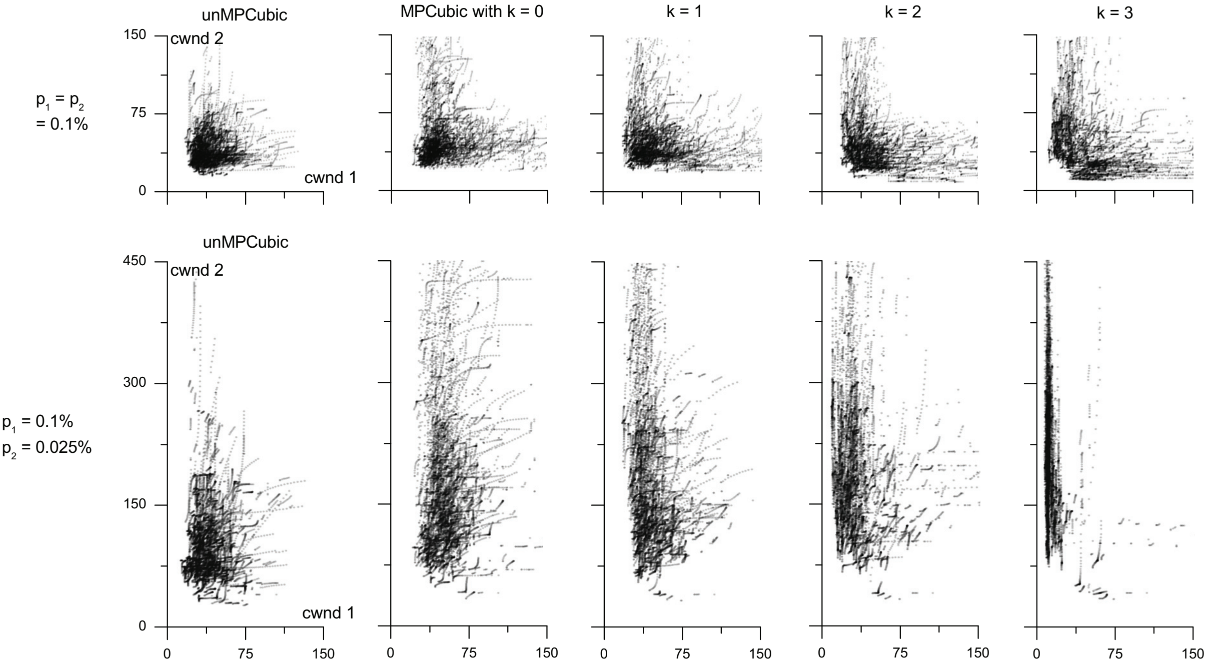


Fig. 1. The evolution of the congestion windows of two-path multipath transport protocols.

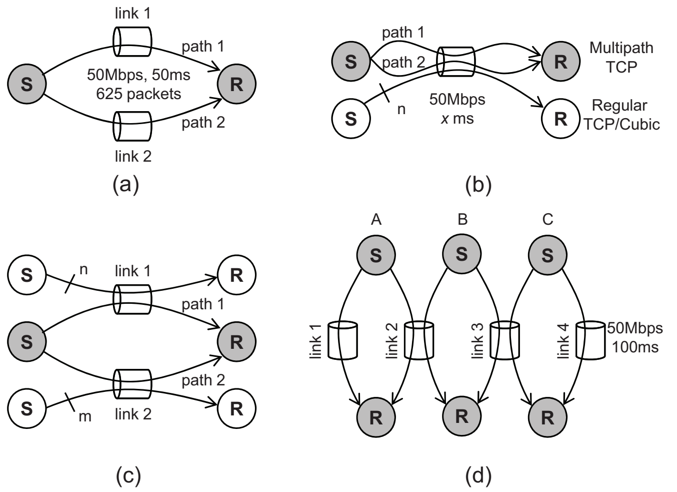


Fig. 2. Simulation scenarios.

window growth cycle of sub-flow in MPCubic equals 2.5 times as long as that of regular Cubic. This result matches the theory above as follows: From (2), and neglecting argument 1 in $\min(.)$, $K_s = K_s^{Cubic}/\delta_s^{1/3}$. If we choose a fixed scale weight of 1/2 (because of using the same $RTTs$ for two paths) for δ_s, $\delta_s = (1/2)^4$. Therefore $K_s = 2^{4/3}K_s^{Cubic}$. Since two paths's parameters are equal, the window evolution of two sub-flows is overlapped.

B. *Fairness*

In this section, we investigate MPCubic's fairness to regular Cubic, TCP (TCP Reno), and MPTCP at common bottleneck.

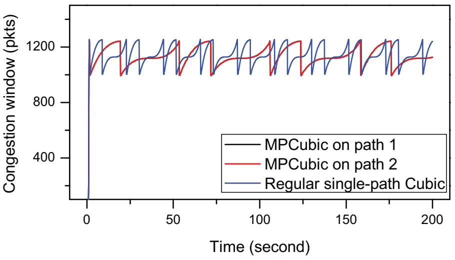
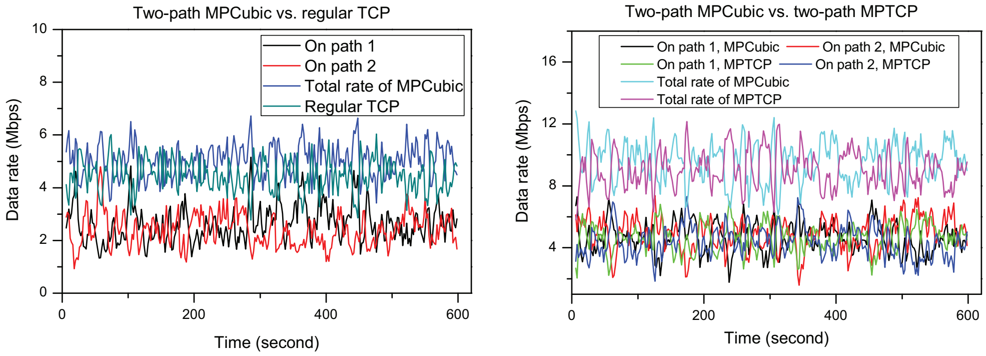


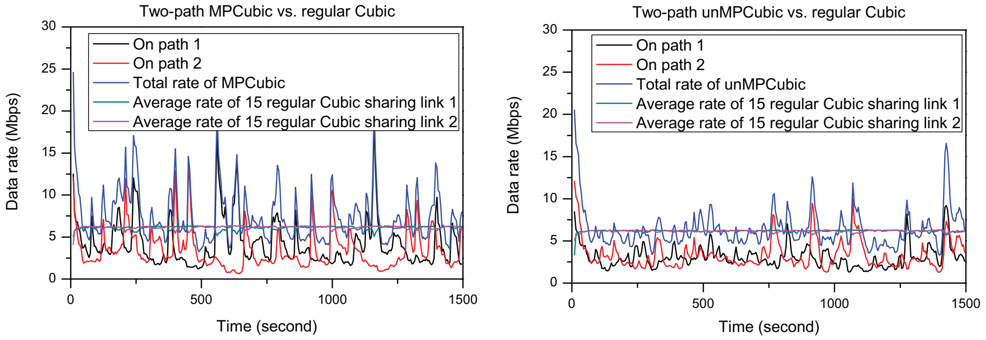
Fig. 3. Congestion window evolution of MPCubic and regular single-path Cubic.

In the first experiment, a two-path MPCubic flow shares with a regular Cubic flow at a link as shown in Fig. 2(b), with link's capacity of 50Mbps and propagation delay of 100ms. The simulation results in Fig. 4 show that two-path MPCubic can be friendly regular Cubic at a bottleneck. The second experiment was run by replacing Cubic by a regular TCP with link's capacity of 10Mbps and propagation delay of 16ms. We choose a short RTT (32ms) such that MPCubic operates in the TCP-friendliness mode as found in [2]. Fig. 5(a) shows that two-path MPCubic is not greedy to capture more bandwidth than the regular TCP. The third experiment shows a two-path MPCubic flow competing against a two-path MPTCP on two separate links with the same parameters as in the second experiment. Fig. 5(b) shows that in the TCP-friendliness region MPCubic can allocate the bandwidth reasonably with regular MPTCP.



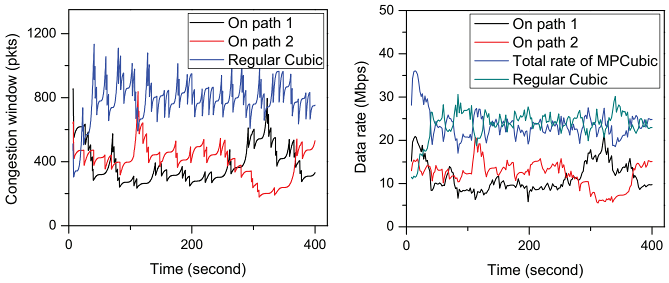
(a) (b)

Fig. 5. (a) Data rate of two-path MPCubic flow vs. regular TCP flow at the shared bottleneck; (b) data rate of two-path MPCubic flow competing against two-path MPTCP flow on two separate links.



(a) (b)

Fig. 6. (a) Data rate of two-path MPCubic flow vs. fifteen Cubic flows on each path; (b) data rate of two-path unMPCubic flow vs. fifteen Cubic flows on each path.



(a) (b)

Fig. 4. (a) Evolution of the congestion windows of a two-path MPCubic flow and a regular Cubic flow sharing at the bottleneck; (b) their data rates.

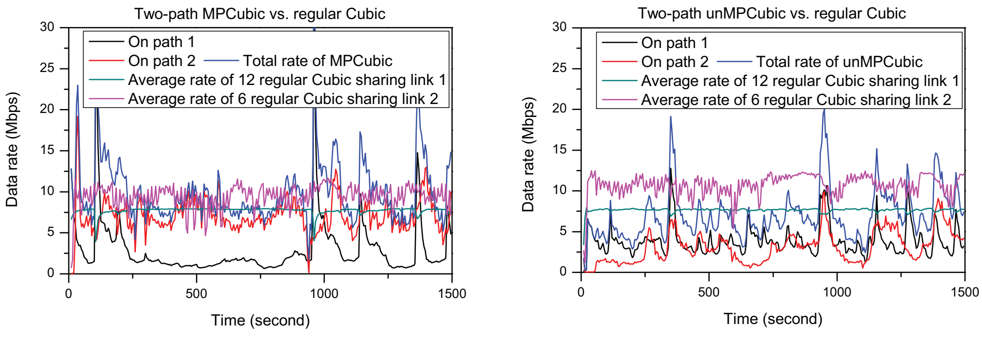
C. Throughput Evaluation

In this section, we demonstrate the improvement of the throughput of MPCubic compared with unMPCubic in a scenario shown in Fig. 2(c), where two links have the same configuration with link's capacity of 100Mbps and propagation delay of 50ms, and the background traffic is generated by fifteen Cubic flows on each link ($n = m = 15$). Fig. 6(a) shows that total throughput of a two-path MPCubic flow is greater than average throughput of fifteen Cubic flows on each path. Two-path unMPCubic flow in Fig. 6(b) gives less fluctuation of the obtained bandwidth, and lower throughput than MPCubic. Therefore, multipath transport protocols can yield the more benefits as the more paths are used.

D. Resource Pooling

In this section, we evaluate resource pooling in terms of throughput and loss rate with the scenario shown in Fig. 2(d). Simulations were run for 400s. Table I shows average throughput of MPCubic and unMPCubic. A perfect resource pooling should produce the equal throughput for each flow. Table I shows that MPCubic appears more fair than unMPCubic, and MPCubic's sum throughput is larger than that of unMPCubic as well. Therefore MPCubic can achieve the throughput improvement. To evaluate the capability of traffic shifting from the congested link, we change link 1's capacity from 50Mbps to 25Mbps. Table II shows that the loss rates always increase as reducing link 1's capacity. For unMPCubic the loss rate at link 1 increases five times because unMPCubic is a uncoordinated congestion controller. Since MPCubic is able to move traffic away from the congested link (link 1), the loss rate is double.

An additional experiment was run on the scenario in Fig. 2(c) with a heterogeneous configuration, where links 1 and



(a)

(b)

Fig. 7. (a) Two-path MPCubic vs. regular Cubic; (b) two-path unMPCubic vs. regular Cubic in a heterogeneous configuration.

TABLE I

THROUGHPUT OF MPCUBIC VS. UNMPCUBIC.

Flow ID	MPCubic	unMPCubic
A	63.7Mbps	64.1Mbps
B	43.7Mbps	40.6Mbps
C	64.2Mbps	63.3Mbps
Sum	171.6Mbps	168Mbps

TABLE II

PACKET LOSS RATE VS. LINK'S CAPACITY.

Link ID	MPCubic	unMPCubic
link 1 = 50Mbps	0.0020%	0.0029%
link 1 = 25Mbps	0.0032%	0.0155%

2 were set to (100Mbps, 40ms) and (80Mbps, 160ms), respectively. Links 1 and 2 are loaded by twelve and six Cubic flows ($n = 12, m = 6$), respectively. Hence link 1 suffers the more congestion. Fig. 7(a) shows that MPCubic produces less aggressive rate on path 1, but more aggressive rate on path 2. Whereas, unMPCubic in Fig. 7(b) gets the total throughput lower than MPCubic because there is no coordination of congestion control between paths.

V. CONCLUSION AND FUTURE WORK

We propose an extended version of regular Cubic for multiple paths over high bandwidth-delay networks, called MPCubic. By establishing coordinated congestion control between paths and constraining fairness to regular single path Cubic/TCP per-path and per-flow, our MPCubic is able to move traffic away from congested paths to lightly load paths, and fairly share to regular Cubic and TCP at common bottleneck, while preserving fairness. Our simulations show that MPCubic can achieve stability, greater throughput, fairness to regular TCP, Cubic and MPTCP, and load-balancing under a variety of network conditions.

We believe our extension technique can generalize for extending current transport protocols for multiple paths. In the future, we will implement MPCubic algorithm in Linux for testbeds, and try to extend other high performance transport protocols such as Compound TCP, FAST TCP.

ACKNOWLEDGMENT

This research was supported by the IT R&D program of KCA(10913-05004: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity). Dr. CS Hong is the corresponding author.

REFERENCES

[1] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," in *IEEE INFOCOM*, 2004.

[2] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," in *ACM SIGOPS Operating System Review*, July 2008.

[3] C. Raiciu, D. Niculescu, M. Bagnulo, and M. Handley, "Opportunistic mobility with multipath TCP," in *Proc. of the MobiArch Workshop*, 2011.

[4] A. Ford, C. Raiciu, and M. Handley, "TCP extensions for multipath operation with multiple addresses," IETF Internet Draft, Mar. 2010.

[5] C. Raiciu, M. Handley, and D. Wischik, "Coupled multipath-aware congestion control," IETF Internet Draft, July 2010.

[6] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," in *ACM SIGCOMM CCR*, Oct. 2008.

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

[8] 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," in *Proc. of the USENIX'04 Annual Technical Conf.*, June 2004.

[9] H. Hseih and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," in *Proc. of the MobiCom' 02 Conf.*, 2002.

[10] J. R. Iyengar, K. C. Shah, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Transactions on Networking*, pp. 951–964, 2006.

[11] L. Magalhaes and R. Kravets, "Transport level mechanisms for bandwidth aggregation on mobile hosts," in *Proceedings of the Ninth International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 165– 171.

[12] 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.

[13] D. Wischik, M. Handley, and C. Raiciu, "Control of multipath TCP and optimization of multipath routing in the Internet," in *Network Control and Optimization*, ser. LNCS. Springer Berlin / Heidelberg, 2009.

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

[15] M. Mathis, J. Mahdavi, S. Floyd, and A.Romanow, "TCP selective acknowledgment options (SACK)," IETF RFC 2018, Oct. 1996.

[16] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," Tech. report, Tech. Rep., 2001.