Alheid, A. A. Y., Kaleshi, D., & Doufexi, A. (2014). Performance Evaluation of MPTCP in Indoor Heterogeneous Networks. In Proceedings of the 2014 First International Conference on Systems Informatics, Modelling and Simulation (SIMS '14). (pp. 213-218). Association for Computing Machinery (ACM). DOI: 10.1109/SIMS.2014.40

University of Bristol - Explore Bristol Research

General rights

Performance Evaluation of MPTCP in Indoor Heterogeneous Networks

Amani Alheid, Dritan Kaleshi, Angela Doufexi
Department of Electrical and Electronic Engineering
University of Bristol
Bristol, UK
{amani.alheid, dritan.kaleshi, a.doufexi}@bristol.ac.uk

Abstract— **This paper studies the effect of out-of-order packets on the MPTCP performance when using different access technologies, modelled to represent path diversity in terms of delay and reliability. The study identifies trade-offs between different CC algorithms in terms of aggregate throughput gain when different packet reordering recovery solutions being implemented. Our analysis shows that TCP-DOOR performs best with both coupling and uncoupling controllers, while DSACK outperforms others under fully coupled. However F-RTO provides better application throughput particular with uncoupled controller and has lower memory requirements compared to others. Adding to that, our observations show that MPTCP suffers from reordering problem and needs a suitable recovery method to achieve optimal aggregate throughput.**

Keywords-multipath TCP; packet reordering; wi-fi; throughput.

I. INTRODUCTION

Computer networks for the home and small business can be build using either wired or wireless technology or probably hybrid. Many products support the 802.11a, 802.11b/g/n, and/or 802.11ac wireless standards collectively known as Wi-Fi technologies. In wireless networks, Wi-Fi ad-hoc mode is a method for wireless nodes or terminals to directly communicate with each other in peer-to-peer fashion without involving central access point. The demand for network efficiency and performance, robustness and reliability is becoming essential. In particular, using more than one network interface, which is already supported by most current devices, for simultaneous data transmission can increase the reliability of the connection as well as the end-to-end throughput.

Whilst traditional transport layer protocols, such as TCP and UDP, only support one IP address at each endpoint in one connection , there is strong interest in the designing of multi-homing protocols from a transport layer perspective based either on TCP or SCTP [1-7]. This paper focuses on MPTCP [1], the extension of the legacy TCP transport layer protocol as a multi-homed protocol proposed by IETF and the only one (based on TCP) that has a real implementation which opens the door for all researchers to study, analyze and improve the behavior of MPTCP protocol widely.

Multipath TCP (MPTCP) is a modification of the classical TCP that allows end-to-end data traffic to be split across multiple paths, whilst maintaining TCP connections at the end points (applications) [1]. The design objectives of MPTCP are to support unmodified applications that use TCP, work over current networks, and to be backward compatibility with regular TCP. Several studies have proposed and analyzed the performance of MPTCP congestion control (CC) algorithms against its goals [8], [9]. They differ in terms of transmission robustness, fairness and path selection stability (flapping), and are discussed more extensively in Section II-B. They all report improvements in throughput measurements compared to the standard TCP. However, sending data through different paths increases the possibility that the order of the packets received at the destination is different from that of the sender (out-of-order (OOO) events). Whilst there exist many causes for packet reordering for a single connection [10] however, when one end-to-end data connection uses more than one path in its transmission then the diversity of path characteristics (both loss and delay) will create OOO events even when the single paths behave ideally.

Although MPTCP is being standardized by IETF, little is understood about how well it performs in dynamic environments such as wireless networks. Most studies have shown improvement in throughput of MTCP compared to single TCP path [11-17]. However, no explicit declaration of packet reordering solutions is implemented. On the other hand, our previous study [18] addressed the MPTCP performance and the reordering problems as the delay variation between two paths increases in an abstract manner. The study showed that when single-path packet reordering algorithms were used both TCP-DOOR and DSACK utilize paths effectively, while the Eifel is not the best choice for throughput maximization. In this paper we focus on the performance of MPTCP over a hybrid network composed of wired and wireless paths.

In this paper we evaluate and compare the behavior of different MPTCP congestion controllers in combination with different packet reordering (PR) recovery solutions. Four of these have been used in this study: D-SACK [19], Eifel [20], TCP-DOOR[21], and F-RTO[22]. The study scenario consists of using nodes with lossless wired interface concurrently with a lossy Wi-Fi interface configured for an indoor environment where the wireless channel is fading in time and frequency with the assumption of Rayleigh channel modeled using Nakagami model. The characteristics of two paths are different mainly in terms of RTTs. Our objective is to study the behavior of the four different reordering solutions when being involved with MPTCP under scenarios of increased retransmission events and reordered received packets due to path diversity.

This paper is organized as follows: Section II presents the MPTCP protocol and the different congestion control algorithms. Section III discusses the four packet reordering solutions proposed for single-path TCP to make it more robust to OOO events and how they have been adopted to MPTCP. In section IV, the experimental results on the performance evaluation of all combinations of CC and PR algorithms for MPTCP for the same network are presented. Section V presents conclusions and future work.

II. MPTCP: MULTIPATH TCP

MPTCP is a modification of the regular TCP that allows single data traffic to be split across multiple paths [1]. One of the main design goals behind MPTCP was to be completely transparent to both the application and the network. The application opens a regular TCP socket which initially starts one regular TCP subflow. More subflows can be added later by any MPTCP end point using the same application socket. Outgoing data is then scheduled according to some implementation management policy and incoming data from all TCP subflows is reordered to maintain the in–order byte–stream abstraction of TCP, as seen by application. For this to work at least one end (preferably both ends) must be multi-homed and both ends must implement the multipath TCP extensions. It has been shown that MPTCP delivers improved network resilience and increased throughput. It can also benefit load balancing at multi-homed servers and data centers [8].

A. Sequence Space

MPTCP protocol uses two levels of sequence spacing: a connection-level sequence number and another sequence number called subflow-level sequence number for each path or subflow (SF). The connection-level sequence is the data sequence number seen by the application. When the MPTCP sender starts transmitting data through different SFs, connection-level data sequence number has to be mapped to the subflow sequence number. Each SF has to send data as a regular TCP connection independently from other SF(s) with its own sequence numbers and cumulative acknowledgments (ACKs). The MPTCP receiver uses the connection-level sequence number to reassemble the data streams coming from different SFs in order to pass them to the application layer in-sequence. Therefore, MPTCP uses a data sequencing mapping (DSM) to convert between the two sequence spacing [1]. The arrival packet is said to be in-sequence if and only if both the subflow-sequence and data-sequence are as expected.

B. Congestion Control

The congestion control (CC) algorithm is the most important part of MPTCP protocol. In the regular TCP protocol, only one congestion window (CWND) exists between the sender and receiver nodes. However the MPTCP sender has a CWND for each subflow to control the local traffic in each path, whilst the MPTCP receiver has a single global receiving window shared between all subflows.

Three major goals for the congestion control have to be satisfied in MPTCP protocol: (1) Improve throughput. (2) Do not harm other flows. (3) Balance congestion [9]. Different CC algorithms have been proposed; Uncoupled (Un-CC), Fully Coupled (FC-CC), and Coupled (Co-CC), and extensive simulation studies have been done to test them against MPTCP goals [9], [23]. Un-CC uses Additive-Increase/Multiple-Decrease (AIMD) congestion control used with regular-TCP in each path independently. The increase and the decrease equations are given by (1) and (2) respectively. However, FC-CC takes total CWND of all links in consideration in order to couple both the increase and decrease cases for each link, as shown by (3) and (4). The Co-CC couples only the increase case for each link as in (5) and keeps the decrease similar to regular-TCP. W_r is CWND of path r, W is the summation of all CWNDs, and α is calculated using (6) [23].

Uncoupled-CC:

$$W_r = Wr + 1/(Wr) \tag{1}$$

$$W_r = \frac{Wr}{2} \tag{2}$$

Fully Coupled-CC:

$$W_r = W_r + \frac{1}{W} \tag{3}$$

$$W_r = max\,(Wr - \frac{W}{2}, 1) \tag{4}$$

Coupled-CC:

$$W_r = W_r + \mathrm{Min}\,(\frac{\alpha}{W}, \frac{1}{Wr}) \tag{5}$$

$$\alpha = W * \frac{\mathrm{Max}\frac{Wr}{(\mathrm{RTT})^2}}{(\Sigma_\mathrm{r}\frac{Wr}{\mathrm{RTT}})^2} \tag{6}$$

Although Co-CC adjusts CWND size for each path taking in consideration RTT measurement, MPTCP cannot saturate link with higher RTT, because OOO data arrival on the receiver endpoint at the connection level causes a bottleneck in data re-sequencing process. Section IV of this report shows that sending data using the best path will be a suitable solution but it limits the aggregate throughput to be no more than the throughput of the best path.

III. PACKET REORDERING SOLUTIONS

A sender generates a traffic stream with an in-order sequence of data packets. For many reasons the ordering of the packets received at the destination may be different from the sender generated one. An out-of-order packet makes the receiver responds with duplicated acknowledgements (dupACK) inducing the sender to infer wrongly a packet loss and then enter congestion control stage unnecessarily, resulting in lower overall end-to-end performance.

In the multipath context, packets arrive OOO because different SFs may have different characteristics, such as end–to–end delay. The OOO arrival of the data packets will create a substantial problem for multipath TCP at reassembling stage at the connection level. When the receiver node receives OOO packets it will store them into OOO buffer waiting for the others. However, when the sender receives dupACKs it will trigger one of the proposed methods for solving reordering in addition to the congestion control selected for the corresponding SF.

Many packet reordering mechanisms have been proposed for single-path TCP as a solution for the OOO problem [10]. Four of them named D-SACK, Eifel, TCP-DOOR and F−RTO will be discussed briefly in this section.

D-SACK is an extension of SACK option for TCP [19] that depends on duplicate selective acknowledgement (D−SACK) to detect segment reordering When a sender finds that it has made a spurious congestion response based on the arrival of a D-SACK it performs "slow start" to increase the current CWND to the stored CWND before congestion avoidance.

Eifel algorithm is proposed to eliminate the retransmission ambiguity and solve the performance problems caused by spurious retransmissions [20]. It uses the TCP timestamp option with each outgoing segment to a destination and its ACKs. When a packet loss is assumed, the sender retransmits the lost segment and stores the timestamp of the first retransmission. Upon receiving the ACK, the sender compares the timestamp of the arrived ACK with the stored one. If it is smaller, then the retransmission was spurious.

TCP-DOOR (Detection of Out-of-Order and Response) has been proposed to improve TCP performance over mobile ad-hoc networks [21]. Once the OOO is detected, then it responds by temporarily disabling the congestion control and instant recovery during congestion avoidance. The sender will keep its state variables constant for a time period and then recover immediately to the state before congestion avoidance action was invoked.

A Forward RTO Recovery (F-RTO) algorithm is a TCP sender method that does not require any TCP options to operate [22]. After retransmitting the first unacknowledged segment triggered by a timeout, the F-RTO algorithm at a TCP sender monitors the incoming acknowledgments to determine whether the timeout was spurious or not and also to decide whether to send new segments or retransmit unacknowledged segments.

In the following section, the packet reordering solutions mentioned previously will be simulated with MPTCP to evaluate their influence on the link utilization and the application throughput.

IV. PERFORMANCE EVALUATION

In this section, we present our simulation results and discuss the path utilization using various packet reordering recovery algorithms mentioned in section III. MPTCP has been simulated using ns-3 [24], [25] and the performance has been evaluated for four different PR solutions namely (D−SACK, Eifel, TCP−DOOR, and F−RTO). The simulated scenarios evaluate the impact of PR on the aggregate throughput (gThroughput) of the protocol for an indoor environment using different access technologies (both wired and wireless links) simultaneously. The wireless link technology used in the simulations is IEEE 802.11g.

A. Simulation setup

The simulated system shown in Fig. 1 assumes an FTP application to transfer a 500MB file under Client/Server architecture. Two nodes are implemented and connected by two different links that represent two possible disjoint paths for MPTCP. One path is presented by a wired network while the second path is wireless. The wired network is set to be lossless link with 6Mbps data rate and 10ms delay. In the case of the wireless link, the downlink PHY of IEEE802.11g standard is implemented [26] for different MCS (Modulation and Coding Scheme). Simulation has been carried out for the case of MCS1 (BPSL ½) which results in raw data rate of 6 Mbps. The IEEE802.11g is operating at 2.6 GHz with fixed bandwidth of 10 MHz. The link characterisation analysis was obtained by considering user terminal locations on an evenly spaced grid between the two endpoints in the range of 150m considering large indoor environment. The transmission power of the user terminal is set to 15dBm, which is a typical value for laptops. An isotropic antenna is assumed in order to provide a generic channel model with no impact from antenna system. The communication channel is generated using the Nakagami [27] propagation model, which models the fading of the wireless channels, and the LogDistance path loss models [28] shown in (7).

$$PL = 10 * m * \log(d/ d_{ref}) + PL_{ref} \qquad (7)$$

Where m is path loss exponent, d is distance in meter, d_{ref} is the reference distance which is assumed 1 in this study, and PL_{ref} is the reference path loss which is set to 46.67 dB. The size of OOO receiver buffer is set to be large enough for all OOO packets so it has no impact on system performance evaluation.

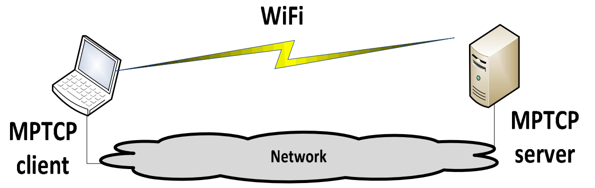

Figure 1. The Simulated Scenario

B. Performance Metrics

In this paper, the following performance metrics are used for the results comparisons and analysis.

1) Out-Of-Order Ratio.

A key performance metric of using PR solution with MPTCP is to measure Out-Of-Order-Ratio (OOO-R) at the receiver side. MPTCP maintains two sequence numbers for each packet: the data sequence for the MPTCP connection and one subflow sequence for each TCP subflow. In-order packets arriving from the same subflow may wait in the

OOO-receive-buffer before their data sequence numbers become in-order – triggered by the (late) arrival of packets from other SFs. OOO-R is measured to be total number of received packets being stored in OOO-buffer over the total number of non-duplicate received packets.

2) Link Utilization.

The link utilization (L-Utilization) can be obtained by observing the SF-CWND. If MPTCP is able to increase the value of CWND then more data can be sent through this SF. The lack of competition in the link from other flows in our scenario makes all bandwidth available to the MPTCP connection. Link Utilization is defined as the ratio of the observed throughput of the link (SF) over its theoretical capacity.

3) Aggregate Throughput.

As our goal is to study the PR impact on the overall performance of MPTCP, we focus on measuring its aggregate throughput (gThroughput). The aggregate throughput is defined by the summation of the throughputs of all available paths for MPTCP connections (SFs). The optimal throughout used in this paper presents the maximum possible throughput that can be achieved by the protocol when consuming all the available bandwidth of the links.

C. Results Analysis

Our baseline performance is the performance of MPTCP protocol without any solution for the reordering problem (NoPR). The impact of four packet reordering solutions of the performance of MPTCP is then compared with the baseline behaviour.

1) MPTCP with NoPR.

The benchmark simulation uses MPTCP with the three CC algorithms (Uncoupled, Fully-Coupled, and Coupled) without using any packet reordering solution while increasing the distance between the client and the server from 10m to 150m.

We observe that the gThroughput of MPTCP protocol using all mentioned CC algorithms is around 4Mbps depending on one SF only to avoid OOO packets arrival at the destination node as shown in Table I. The Reorder Ratio is very low and it does not exceed more than 2% in average for all simulated scenarios because most of the packets arrive in-sequence when one of the SF dominates over the other.

Although the first goal of the MPTCP protocol design is satisfied, the gThroughput is not optimal as the capacity of SF-1 is not used. This is because when the transmission starts one of the subflows suffers from late packet arrival and cannot recover from reordering problem.

2) MPTCP with PR Solutions.

In this part of our evaluations, the MPTCP is simulated with four mechanisms proposed for single-TCP to recover from OOO using the same topology of Fig.1. This section presents a complete analysis when the distance between two endpoints equal to 10m, as a typical set of results. The performance analysis of these PR solutions is compared with the baseline evaluations presented previously.

TABLE I. AVERAGE gTHROUGHPUT WHEN THE DISTANCE BETWEEN 2 ENDPOINTS INCREASES FROM 10M TO 150M.

Congestion Controller	**Average gThroughput (Mbps)**				
	NoPR	***D-SACK***	***Eifel***	***F-RTO***	***TCP-DOOR***
Uncoupled	4.13	6.88	6.33	8.11	7.86
Fully Coupled	4.13	8.35	6.82	6.75	7.89
Coupled	4.15	6.08	6.35	6.72	7.81

From our observations, PR solutions increase OOO-R up to 30-fold compared to original MPTCP. OOO-R reaches 48 % in maximum with TCP-DOOR using coupling methods and 31% on average with D-SACK. As in Table II, the increase of the OOO-R indicates that the sender, with the help of a PR solution, is able to realise the late arrival of packets and therefore rolls back CWND to its state exactly before retransmission was triggered and continue sending more data. The OOO-buffer occupation increases as more packets are stored waiting for their data-sequence to be in-order. Table III shows also link utilization for both SFs and gThroughput obtained in all studied cases; L-Utilisation is the proportion of maximum single-path capacity used by the respective MPTCP subflow.

TABLE II. OOO-R AND MAXIMUM OOO BUFFER SIZE OCCUPIED BY DIFFERENT PR SOLUTIONS AND MPTCP CCS

Reordering Solution	**Uncoupled**		**Fully Coupled**		**Coupled**	
	Reorder Ration (%)	***OOO-Buffer (KB)***	***Reorder Ration (%)***	***OOO-Buffer (KB)***	***Reorder Ration (%)***	***OOO-Buffer (KB)***
NoPR	2.21	5.6	1.58	5.6	1.60	5.6
D-SACK	31.08	18.2	1.60	210.0	17.04	168.0
Eifel	24.8	144.2	4.86	79.8	17.57	72.8
F-RTO	19.79	39.2	14.25	50.4	16.35	50.4
TCP-DOOR	43.26	331.8	48.29	190.4	48.41	145.6

The results show that packet reordering solutions increase gThroughput as MPTCP now utilises all available paths. Aggregate throughput improves by at least 70% in all cases, and more than 100% when using TCP-DOOR. Also it is worth noting that the improvement varies across different combinations of PR and congestion coupling algorithms. For example, using D-SACK improves the MPTCP gThroughput by 80-100% against NoPR case with FC-CC algorithms, whilst F-RTO combined with Un-CC gives 70−100% improvement. Eifel performs less than the others by providing about 30% improvement for all scenarios.

As expected, TCP-DOOR has a significant impact on the gThroughput and link utilization as it suspends the congestion response for a certain time period upon detecting a spurious retransmission, specifically targeting wireless links where variation in RTT could be due to link dynamics rather than congestion. The gThroughput obtained from all PR solutions with all CC are presented in Table III.

TABLE III. MPTCP gTHROUGHPUT AND L-UTILIZATION COMPARISONS BETWEEN DIFFERENT PR SOLUTIONS AND MPTCP CCs

Reordering Solution	Uncoupled			Fully Coupled			Coupled		
	L-Utilization (%)		*gThroughput (Mbps)*	*L-Utilization (%)*		*gThroughput (Mbps)*	*L-Utilization (%)*		*gThroughput (Mbps)*
	Wi-Fi	*Wired*		*Wi-Fi*	*Wired*		*Wi-Fi*	*Wired*	
NoPR	68.11	2.15	4.22	68.10	1.88	4.20	68.10	1.94	4.20
D-SACK	66.06	48.49	6.87	52.68	94.58	8.84	67.43	62.35	7.79
Eifel	66.65	39.28	6.36	66.55	49.49	6.96	66.57	26.51	5.59
F-RTO	66.08	68.76	8.09	60.07	55.25	6.92	60.41	50.49	6.65
TCP-DOOR	63.44	85.41	8.93	68.43	63.29	7.90	68.39	63.46	7.91

By examining the OOO-Buffer, F-RTO outperforms the other combinations in terms of memory requirements. F–RTO occupies less memory space (63KB in maximum) as compared to other PR solutions; this makes F–RTO preferable to others in terms of memory utilization. A typical set of results are shown in Table II where in general, TCP-DOOR consumes more memory than the others because it allows the sender to continue sending more packets even if congestion is detected by 3 duplicated ACKs.

In order to give an overview of all results under different distances between sender and receiver nodes, the aggregate throughputs of MPTCP with all PR solutions being simulated as a function of distance are shown in Fig.3, Fig.4, and Fig.5 under un-CC, FC-CC, and Co-CC labels respectively.

The gThroughput figures show that the original MPTCP does not reach the optimal throughput without implementing a solution for the reordering problem. The base line measurement of gThroughput of MPTCP under three congestion control methods did not exceed 4 Mbps even with short distacne between two endpoints. All PR solutions give between 50%-100% end-to-end throughput improvement over the NoPR MPTCP, depending on the pairing between packet reordering solution and the congestion control algorithm. Both TCP-DOOR and F-RTO are more tolerant to wireless link variations (represented as SNR changes due to distance variation). In addition, TCP–DOOR gives the best performance with both coupling and uncoupling controllers, with D–SACK outperforms others under FC-CC. If memory requirements due to OOO bufer are the main metric, F-RTO provides better application throughput specially with Un-CC and requires less memory utilization as compared to others.

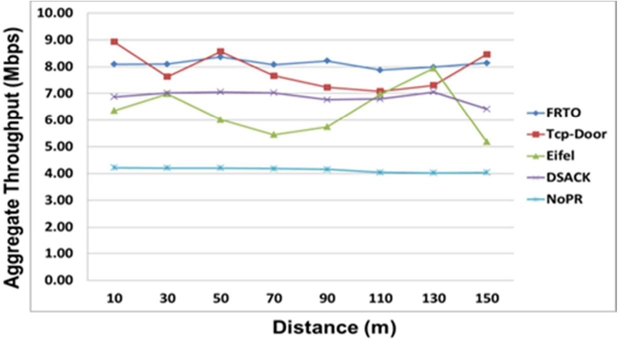

Figure 2. The gThroughput of MPTCP with Uncoupled-CC and various packet reorder solutions as distance between two endpoints increases

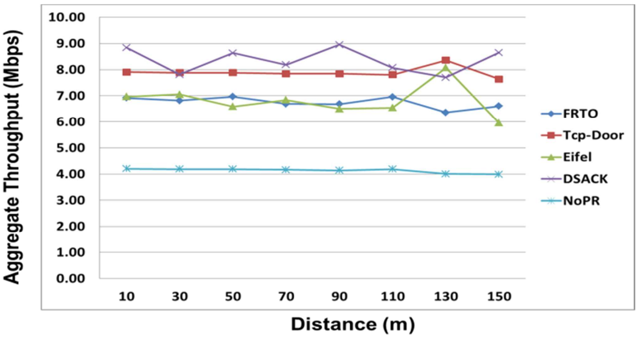

Figure 3. The gThroughput of MPTCP with Fully Coupled-CC and various packet reorder solutions as distance between two endpoints increases

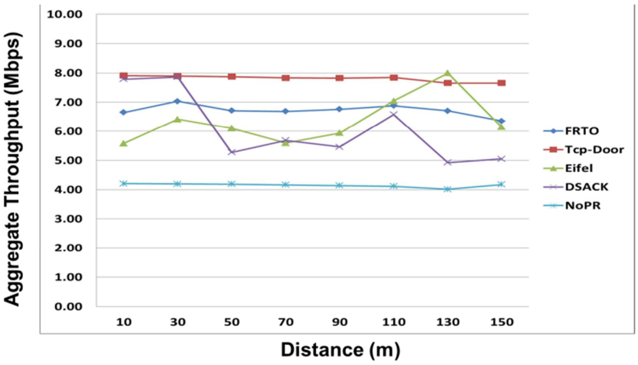

Figure 4. Figure 1The gThroughput of MPTCP with Coupled-CC and various packet reorder solutions as distance between two endpoints increases.

V. CONCLUSION

This paper evaluates some of packet reordering solutions proposed to solve the packet reordering problem in single–path TCP in the context of multipath TCP and compares them when run in conjunction with different congestion controllers under a combined wired and indoor wireless scenario. This study presents results of the performance of MPTCP with four TCP packet reordering solutions, namely D-SACK, Eifel, TCP-DOOR, and F-RTO, and benchmarks them against the performance of MPTCP without any packet reordering recovery methods. The results show that the throughput gain of MPTCP needs a proper solution for the OOO problems. Whilst MPTCP gives a good network resilient and provide load balancing with a coupled congestion controller, it is not better than the single TCP without solving OOO problems which has been produced by the variation of link characteristics.

The results clearly show that the packet reordering solutions bring a substantial performance improvement for MPTCP by increasing the aggregate throughput as well as the path utilization. MPTCP should use F-RTO as a PR solution if memory is a constraint.

ACKNOWLEDGMENT

This research is supported in part by the Public Authority for Applied Educational and Training (PAAET), Kuwait.

REFERENCES

[1] A. Ford, C. Raiciu., M. Handley, and O. Bonaventure. "TCP Extensions for Multipath Operation with Multiple Addresses". *RFC 6824,* 2013

[2] A. Adu-Al, T. Saadawi, and M. Lee, "LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol," *Comput. Commun.*, vol. 27, no. 10, pp. 1012–1024, 2004.

[3] H. Hsieh.and R. Sivakumar, "pTCP: an end-to-end transport layer protocol for striped connections," *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on* , pp.24,33, 12-15 Nov. 2002 "

[4] R. Stewart, Q. Xie, et al., Stream Control Transmission Protocol, *RFC 2960, IETF,* 2000

[5] D. Sarkar, "A Concurrent Multipath TCP and Its Markov Model," *Communications, 2006. ICC '06. IEEE International Conference on* , vol.2, no., pp.615,620, June 2006

[6] Y. Dong, D. Wang, N. Pissinou, and J. Wang. "Multi-path load balancing in transport layer." In *Next Generation Internet Networks, 3rd EuroNGI Conference on*, pp. 135-142. IEEE, 2007.

[7] Y. Hasegawa, I. Yamaguchi, T. Hama, H. Shimonishi, and T. Murase. "Improved data distribution for multipath tcp communication". *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE,* vol.1,pp. 5-pp., IEEE, 2005

[8] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley "Design, implementation and evaluation of congestion control for multipath TCP" *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, pp.8-8 , 2011

[9] C. Raiciu, D. Wischik, and M. Handley. "Practical congestion control for multipath transport protocols." *University College of London Technical Report* (2009).

[10] K. Leung, V. Li. & D. Yang "An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges". *Parallel and Distributed Systems, IEEE Transactions on, IEEE, vol. 18*, pp. 522-535, 2007.

[11] C. Paasch, R. Khalili, and O. Bonaventure, "On the benefits of applying experimental design to improve multipath TCP," presented at the Proceedings of the ninth ACM conference on Emerging networking experiments and technologies, Santa Barbara, California, USA, 2013.

[12] Y. Chen, Y. Lim, R. J. Gibbens, E. Nahum, R. Khalili, and D. Towsley, "A Measurement-based Study of Multipath TCP Performance over Wireless Networks," in ACM IMC, 2013.

[13] N. Sinh Chung, Z. Xiaofei, N. Thi-Mai-Trang, and G. Pujolle, "Evaluation of throughput optimization and load sharing of multipath TCP in heterogeneous networks," in Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on, 2011, pp. 1-5.

[14] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, "Opportunistic mobility with multipath TCP," presented at the Proceedings of the sixth international workshop on MobiArch, Bethesda, Maryland, USA, 2011.

[15] Z. U. Shamszaman, S. S. Ara, and C. Ilyoung, "Feasibility considerations of multipath TCP in dealing with big data application," in Information Networking (ICOIN), 2013 International Conference on, 2013, pp. 708-713.

[16] K. Han Ah, O. Bong-hwan, and L. Jaiyong, "Improvement of MPTCP Performance in heterogeneous network using packet scheduling mechanism," in Communications (APCC), 2012 18th Asia-Pacific Conference on, 2012, pp. 842-847.

[17] C. Shengyang, Y. Zhenhui, and G. M. Muntean, "An energy-aware multipath-TCP-based content delivery scheme in heterogeneous wireless networks," in Wireless Communications and Networking Conference (WCNC), 2013 IEEE, 2013, pp. 1291-1296.

[18] A. Alheid, D. Kaleshi & A. Doufexi "An Analysis Impact of Out-Of-Order Recovery Algorithms on MPTCP Throughput". in Advanced Information Networking and Applications , 2014 in press.

[19] S. Floyd, J. Mahdavi, M. Podolsky, and M. Mathis "An extension to the selective acknowledgement (SACK) option for TCP". *RFC 2883,* 2000.

[20] R. Ludwig, and H. Katz "The Eifel algorithm: making TCP robust against spurious retransmissions". *ACM SIGCOMM Computer Communication Review, ACM,* vol. 30, pp.30-36, 2000.

[21] F. Wang and Y. Zhang "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response". *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing,* pp. 217-225, 2002.

[22] P. Sarolahti, M. Kojo, K. Yamamoto, and M. Hata "Forward RTO-recovery (F-RTO): An algorithm for detecting spurious retransmission timeouts with TCP", *RFC5682, IETF,* 2009

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

[24] (2012) ns-3 website. [Online]. Available: http://www.nsnam.org/.

[25] B. Chihani and D. Collange, "A Multipath TCP Model for ns-3 simulator", *Workshop on ns-3 held in conjunction with SIMUTools 2011*, Spain, 2011.

[26]. B. Mitchell, "Wireless Standards - 802.11b 802.11a 802.11g and 802.11n," *About.com.* 2012.

[27] G. Karagiannidis, N. Sagias, and P. Mathiopoulos, "The N* Nakagami fading channel model," *Wireless Communication Systems, 2005. 2nd International Symposium on* , vol., no., pp.185,189, 5-7 Sept. 2005

[28] M. Hidayab, A. Ali, and K. Azmi, "Wifi signal propagation at 2.4 GHz," *Microwave Conference, 2009. APMC 2009. Asia Pacific* , pp.528,531, Dec. 2009