OPEN ACCESS

University of
BRISTOL

Alheid, A., Doufexi, A., & Kaleshi, D. (2016). A Study on MPTCP for Tolerating Packet Reordering and Path Heterogeneity in Wireless Networks. In Wireless Days 2016, Toulouse. DOI: 10.1109/WD.2016.7461454

Peer reviewed version

Link to published version (if available):
10.1109/WD.2016.7461454

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

A Study on MPTCP for Tolerating Packet Reordering and Path Heterogeneity in Wireless Networks

Amani Alheid, Angela Doufexi
Department of Electrical and Electronic Engineering
University of Bristol
Bristol, UK
(amani.alheid, a.doufexi)@bristol.ac.uk

Dritan Kaleshi
5G Fellow at
Digital Catapult
London, UK
Dritan.Kaleshi@cde.catapult.org.uk

Abstract— **As a great step toward the 5G network that will comprise the set of revolutionary technologies designed to cope with the vast increase in demand for faster connectivity, speedier downloads and more reliable communication, Multipath Transmission Control Protocol (MPTCP) promises higher bandwidth and higher resilience against network path failures by allowing multiple paths between two devices to be pooled and appear to the application as a single end-to-end transport connection. This paper studies the effect of out-of-order packets on the performance of MPTCP when at least one of the paths uses different wireless access technologies. The study identifies trade-offs between different congestion control algorithms in terms of aggregate throughput gain when different packet reordering recovery solutions are implemented. The paper also suggests the best reordering solution for MPTCP for different wireless network conditions in order to maximize link utilization. The results show that MPTCP suffers from a reordering problem and the TCP-DOOR is able to maximize path utilization and provides a substantial improvement in terms of aggregate throughput for symmetrical links whilst D-SACK is preferable for asymmetrical links.**

Keywords— *MPTCP; Packet Reorder; DSACK; Eifel; F-RTO; TCP-DOOR; Congestion Control; Wi-Fi.*

I. INTRODUCTION

Once a decade, a new generation of mobile network technology comes along, starting in the 1980s, where the first mobile networks appeared, until 2010, where the LTE began rolling out. The forthcoming mobile network technology is 5G. 5G has a vast ambiguity because it is largely a concept at this point, and the overall standards have not yet been defined. But it is seeking to achieve some key goals like faster data speeds and ultra-low latency. And here is the idea behind the data offloading and the multipath transmission protocols. Several multipath transmission proposals have been published since 2002 [1-5]. The key motivation behind multipath protocols is the trend toward providing trusted and reliable connectivity in the future Internet. The research in this context focusses on faster downloads, lower data transfer costs and seamless switching between different interfaces, particularly the wireless ones such as Wi-Fi and cellular networks. One such proposal is "Super pipe," which offers reliable connectivity in highly mobile networks by aggregating all available data channels to provide faster and more reliable data transfer [6]. Another such proposal is "Download booster", a proposal by Samsung that significantly boosts the download speeds by taking advantage of Wi-Fi and LTE networks at the same time [7].

One of the main reasons for sub-optimal performance of single or multipath transport protocols, particularly in terms of aggregate throughput, is that of packet reordering [8, 9]. In conventional single-path TCP, the reordering problem comes from packet-level multipath routing, route fluttering, inherent parallelism in modern high-speed routers, link layer retransmission, and router forwarding lulls [10]. In addition to these, in multipath transport context, the reordering comes from the heterogeneity of paths likes the end-to-end delay and the data transmission rate.

This paper analyses the impact of reordering recovery protocols on the throughput gain and path utilization of the multipath TCP (MPTCP), which currently is the most relevant solution at transport layer for multipath transmission in the Internet. Whilst the existing studies on MPTCP show throughput improvement when using MPTCP compared with a single-path TCP, they do not analyse or compare the impact of the reordering solutions. The study presented here examines the influence of the packet reordering on the behaviour of MPTCP extensively and particularly when at least one of the paths ends in a wireless link. It compares the performance of the end-to-end connection for different congestion control (CC) algorithms implemented at subflow-level and different types of packet reordering (PR) solutions integrated with MPTCP at the connection-level. The study assumes MPTCP operating over paths that include two Wi-Fi links. As this study looks at wireless networks as lossy networks with varying link reliability and stability, the results and conclusions provided by this paper can be generalized for any wireless lossy links.

This paper is organized as follows: Section II presents in detail the MPTCP, the different congestion control algorithms, the reordering problems and the recovery mechanisms. Section III describes the system model setup and parameters, simulated scenarios, and performance evaluation metrics. Section IV presents the analysis of the results of all combinations of CC and PR recovery algorithms for MPTCP for the different simulated wireless scenarios, followed by a detailed discussion in Section V. Section VI presents the conclusions of this study.

II. MULTIPATH TCP

MPTCP is a modification of the regular TCP that allows single end-to-end connection data traffic to be split across multiple TCP paths [1]. One of the main design goals behind the 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 (SF). More SFs can be added later by any MPTCP end point using the same application socket. Outgoing data are then scheduled according to some implementation management policy, and incoming data from all SFs are reordered to maintain the in-order byte-stream abstraction of TCP, as seen by the application [11]. For this to work, at least one end (preferably both ends) must have at least two IP addresses, and both ends must implement the MPTCP extensions.

A. Sequence Space in MPTCP

MPTCP uses two levels of sequence spacing, a connection-level sequence number, which is also called data-level sequence number, that is used by the TCP socket and seen by the application, and another sequence number called SF-level sequence number used independently for each SF or each physical path; MPTCP uses data sequencing mapping (DSM) to convert between the two sequence spacing [1]. Since the sender is able to send data through more than one interface, it is very likely that the received packets reach the destination in different order than the sending order, especially when the links have different characteristics (i.e., latency). In this case, the receiver endpoint has to store out-of-order (OOO) packets into OOO buffer before sending them to the receiver buffer which stores all in order packets that are ready to be sent to the application. The arrival packet is said to be in sequence if and only if both sequence numbers (data-level and SF-level) are in sequence.

Figure 1 explains how the MPTCP receiver node examines the newly arrived packet that is not rejected to decide whether to save it in the receiver buffer (in-order packet) or in the OOO buffer (OOO packet). The receiver first checks the SF sequence and then the connection sequence. If the SF sequence number of the received packet (SF_RecSeq) is equal to the expected SF sequence number (SF_ExpectSeq) and the connection (or data) sequence number (D_RecSeq) is equal to the expected data sequence number (D_ExpSeq), then the packet is considered to be in sequence. If either of the sequence numbers is greater than the expected, the received packet is considered to be OOO.

B. Congestion Control in MPTCP

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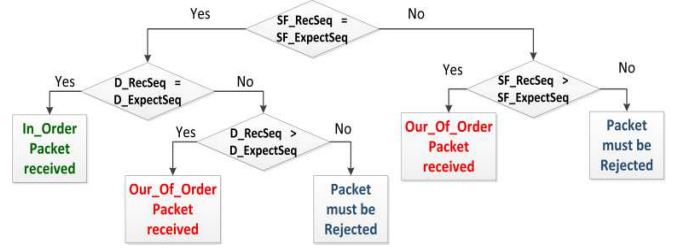

Fig. 1. Packet classification at MPTCP receiver node

Different CC algorithms have been proposed: Uncoupled (Un-CC), Fully Coupled (FC-CC), and Coupled (Co-CC). Extensive simulation studies have been undertaken to test them against the MPTCP design goals [12, 13]. From our previous studies [14, 15], it is shown that MPTCP cannot saturate a link with higher round-trip time (RTT), because OOO data arrival on the receiver endpoint at the connection level causes a bottleneck in the data re-sequencing process.

C. Packet Reordering in MPTCP

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 order. An OOO packet makes the receiver respond with duplicated acknowledgements (DUPACK), inducing the sender to infer erroneously a packet loss and unnecessarily enter the congestion control (CC) stage, resulting in lower overall end-to-end performance.

In a multipath context, packets arrive OOO because different SFs routinely have different characteristics (i.e., end-to-end delay). The OOO arrival of the data packets will create a substantial problem for multipath TCP while reassembling them at the connection-level. When the receiver node receives OOO packets, it will store them into the OOO buffer and wait for the sequentially preceding ones in order to deliver the in-order byte-stream TCP service abstraction to the application. However, when the sender receives DUPACK, it will trigger one of the proposed methods for solving reordering, if enabled, in addition to the CC selected for the corresponding SF. MPTCP encounters a bottleneck in the data-reordering process at a receiver endpoint, and the receiver needs a significant receiving buffer to save OOO packets coming from different SFs [11, 16] in order to achieve the desired performance.

The reordering of the packets is a significant problem even for single-path TCP connections, and several mechanisms have been proposed for single-path TCP to resolve it. Four of them (*D-SACK* [17], *Eifel* [18], *TCP-DOOR* [19] and *F-RTO* [20]) are used and compared in this study in order to evaluate their influence on the link utilization and the application throughput of MPTCP and to estimate the required amount of OOO buffer size. Note that *NoPR* or *MPTCP-NoPR* refers to MPTCP without any PR solutions.

III. System Setup and Performance Metrics

A. System Level simulation

The performance of MPTCP is evaluated for a network where each path uses a wireless link as shown in Fig. 2. The network simulator used is NS-3 with MPTCP module [21, 22]. The system parameters for the wireless interfaces used in the simulations are shown in Table I. The size of OOO receiver buffer is set to be large enough for all OOO packets so it has no impact on system performance (10 MB). The backbone network is set to 100 Mbps data rate and 10 ms delay. The simulation consists of transferring a single large file from an FTP server, where all packets are of equal length (1400 B) and the simulation time is set to 50 sec. The distance between the mobile node and the Wi-Fi access point (AP) is set initially to 10 m and then increased to 100 m in 10 m steps.

TABLE I. SYSTEM PARAMETERS FOR WIRELESS INTERFACES

System Parameters	Wireless Interface I	Wireless Interface II
Standard	IEEE802.11g	IEEE802.11b
Modulation	BPSK1/2 & 64QAM3/4	HR/DSSS
Data Rate	6Mbps & 54Mbps	2Mbps
Antenna	Isotropic	
Fading Channel	Nakagami [23]	
Path Loss Module	LogDistance	

The LogDistance path loss model used is shown in (1).

$$PL = 10 * m * log (d / dref) + PLref \qquad (1)$$

Where m is the path loss exponent, d is the distance in metres, $dref$ is the reference distance which is assumed to be 1 in this study, and $PLref$ is the reference path loss which is set to 46.67 dB.
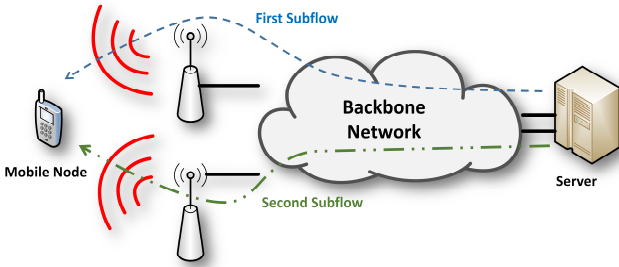
B. Network Topology and Simulated Scenarios

Fig. 2. Network topology for MPTCP wireless scenarios

The study considers two main cases, where each case assumes different link physical (PHY) data rates in the access network; backbone data rate is kept unchanged. In the first case, for reasons of completeness and benchmarking, the performance of the system when the wireless interfaces use the same technology and data rate consists of two wireless networks that have equal PHY data rates as in scenarios 1 and 2. The second case (more realistic) is looking at the performance of the system when the two paths have different characteristics, as shown in in scenarios 3 and 4. The simulated scenarios are presented in Table II.

TABLE II. SIMULATED SCENARIOS

	First SF		Second SF	
	Wireless standard	PHY data rate	Wireless standard	PHY data rate
Similar Wireless Links				
Scenario 1	11g	54 Mbps	11g	54 Mbps
Scenario 2	11b	2 Mbps	11b	2 Mbps
Dissimilar Wireless Links				
Scenario 3	11g	54 Mbps	11b	2 Mbps
Scenario 4	11g	6 Mbps	11b	2 Mbps

C. Performance Metrics

The following performance metrics are used here for the results comparisons and analysis.

1) Out-Of-Order Ratio.

Out of order ratio (OOO-R), which is calculated at the receiver side, is the ratio of the total number of received packets being stored in the OOO buffer to the total number of non-duplicate received packets.

2) Out-Of-Order Buffer Size.

If an OOO packet arrives at the destination, then the packet will be stored in a buffer awaiting the late or lost packets to arrive. The OOO-buffer-size is used to measure the maximum amount of memory required by different PR recovery methods, and it is defined as the maximum number of packets stored in the OOO buffer during simulation time.

3) Aggregate Throughput.

The gThroughput is defined as the sum of the throughputs of all SFs used for MPTCP connections.

IV. Results Analysis

The results are presented for each of the two cases discussed in the previous Section (Case 1: Scenarios 1 and 2, MPTCP performance over similar paths, and Case 2: Scenarios 3 and 4, MPTCP performance using paths with different characteristics.). Due to the huge set of results, the tables only present the results achieved at 10 m distance between client and AP, which corresponds to the highest performance in terms of gThroughput. The boxplot figure is used to depict the gThroughput range achieved through all simulated distances (from 10 to 100 m), where each box represents five main values of the measured throughputs (minimum, first quartile, median, third quartile, and maximum). For completeness of discussion, it is worth mentioning that the performance of single TCP over Wi-Fi is also evaluated (15Mbps is the maximum throughput can TCP achieve over a Wi-Fi link (11g) in NS-3), however the results have not been presented but will be referred to for comparison purposes.

A. Similar Wireless Links

In Scenario 1, TCP-DOOR is the best in terms of gThroughput (up to 25 Mbps regardless of the CC of the protocol), and this is a 66% increase compared to the conventional MPTCP (NoPR). D-SACK with FC-CC approaches TCP-DOOR performance in terms of gThroughput and provides less gThroughput gain (33%) with other CCs. This is applicable for distances of less than 70 m between the

mobile device and the AP because the performance of this Wi-Fi network becomes worse after 70 m.

In terms of out-of-order performance, OOO-R basically increases as the distance between the client and the AP increases due to deterioration of link (lower signal-to-interface-plus-noise ratio (SINR)). Consequently, the delay, retransmissions, and timeout events increase. In this scenario, OOO-R for both TCP-DOOR and D-SACK are larger than the others particularly for distances of less than 70 m (as shown in Table III). This is because both of them are capable of utilizing both paths, which increases the occurrence of OOO events, as well increases in gThroughput compared to the conventional MPTCP. In this scenario, the main factor that impacts the performance of the MPTCP is the PR solution used and not CC strategy, which makes for a very interesting observation, as shown in Fig. 3. The boxplot figure shows the gThroughput range (minimum, first quartile, median, third quartile, and maximum) achieved through all simulated distances. In terms of memory requirements for processing the OOO events, D-SACK requires less memory space than the other PR methods. The OOO-buffer-size for the D-SACK under FC-CC is on average 208 KB smaller than that of TCP-DOOR and 420 KB smaller under other CCs; although, they are providing a comparable gThroughput. Table III and Table IV summarizes the OOO-R and OOO-buffer-size respectively for all scenarios at distance 10 m as a typical set of results.

In Scenario 2, the access network uses IEEE 802.11b standard operating at 2 Mbps data rate. From Fig. 4, it can be observed that MPTCP-NoPR is still unable to utilize both links while TCP-DOOR, D-SACK, and F-RTO can, and it causes an improvement in gThroughput of up to 42% when used with Un-CC. Eifel does not provide any noticeable improvement. The results are comparable with those for Scenario 1 with the exception of F-RTO which, contrary to Scenario 1, performs well in this scenario. Under Un-CC and for all distances, all PR solutions including NoPR provide the same gThroughput with very small variations while the gThroughput under other CCs decreases after 70 m distance (when link quality (SINR) deteriorates). The gThroughput achieved by all PR solutions under Co-CC is shown in Fig.4 as a representative result for all other CCs. Whilst D-SACK and F-RTO come equal second in their gThroughput performance, they require less than half of the memory space of the TCP-DOOR as depicted in Table IV.

B. Dissimilar Wireless Links

This section analyses the impact of the PHY data rate variation between different paths on MPTCP performance for Scenarios 3 and 4, i.e., a network with two dissimilar wireless links, differing mainly in their PHY data rates. Two PHY data rates have been implemented for 11g which are 54 Mbps and 6 Mbps, whilst 2 Mbps is used for 11b standard.

Under Scenario 3, D-SACK provides the largest gThroughput compared to others and regardless of CCs. It achieves up to 15 Mbps when the SINR of the Wi-Fi network is high and particularly for distances less than 50 m. Beyond this distance, the gThroughput decreases as a function of distance (as expected due to link SINR deteriorating). The results show that D-SACK utilizes the (faster) 11g link only,

whilst the conventional MPTCP uses 11b and provides the maximum throughput that can be achieved over this link.

(a) The gThroughput of MPTCP and Un-CC against PR solutions

(b) The gThroughput of MPTCP and Co-CC against PR solutions

(c) The gThroughput of MPTCP and FC-CC against PR solutions
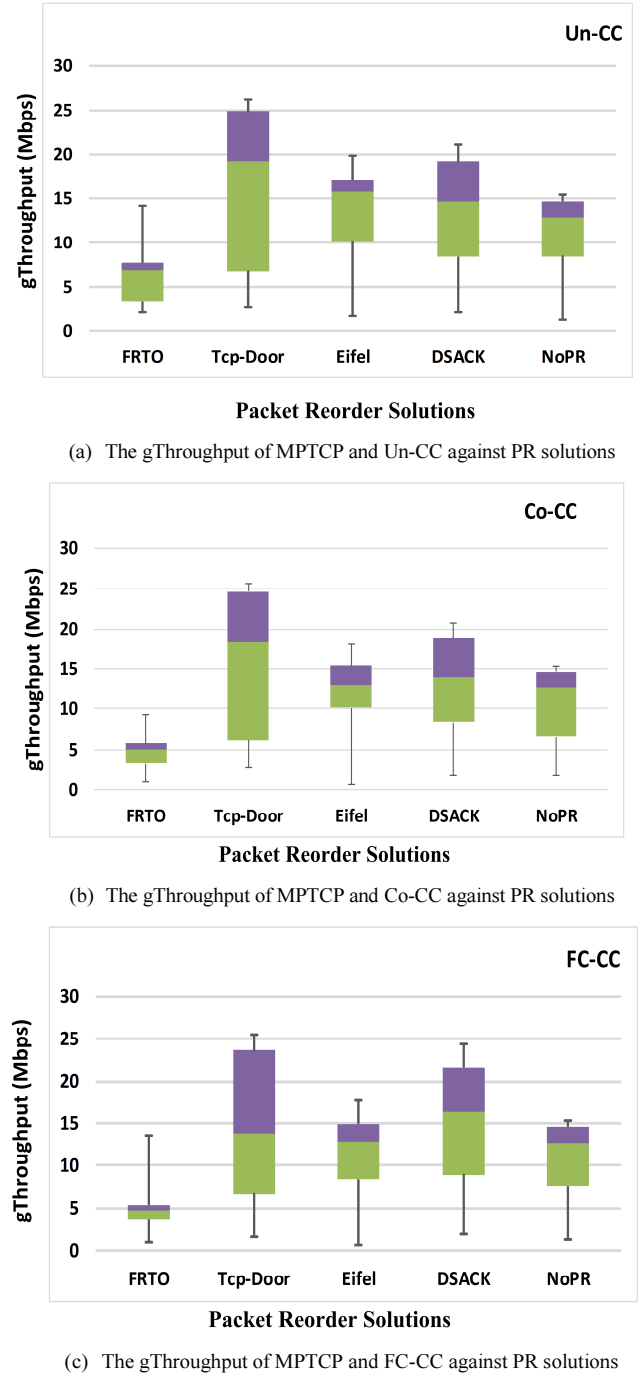
Fig. 3. Scenario 1 gThroughput against PR solutions

Thus, NoPR and Eifel does not exceed 2 Mbps in gThroughput during all simulated distances and for all CCs. F-RTO and TCP-DOOR provide higher gThroughput than NoPR and utilize both links by transmitting through the two SFs while the transmissions of the others PRs is dominating through one SF. On the other hand, their gThroughputs do not

reach the throughput of the best path available (higher data rate). The gThroughput results for Co-CC are shown in Fig. 5 as representative results (FC-CC and Un-CC are similar).
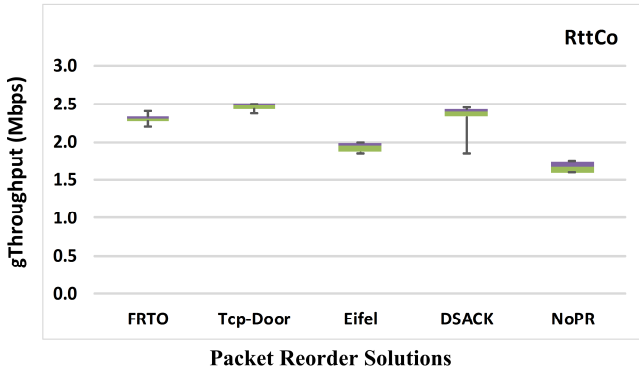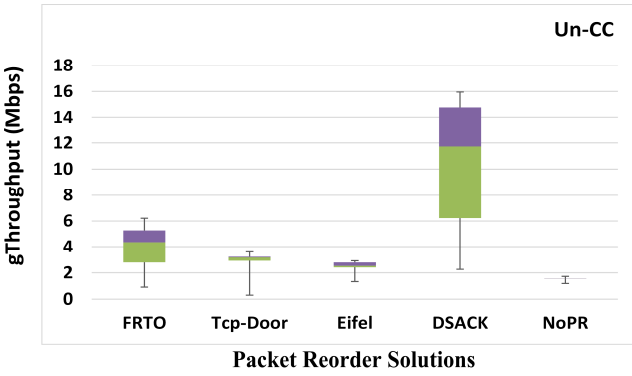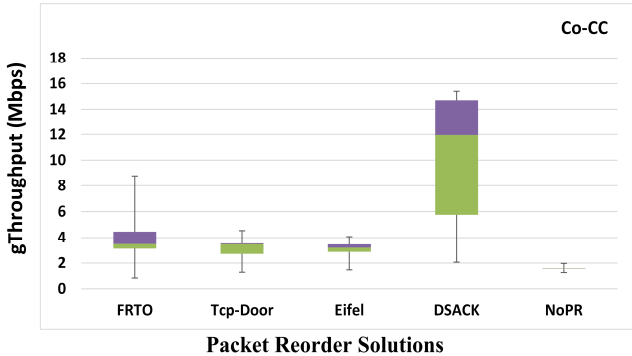

Fig. 4. Scenario 2 gThroughput against PR solutions with Co-CC

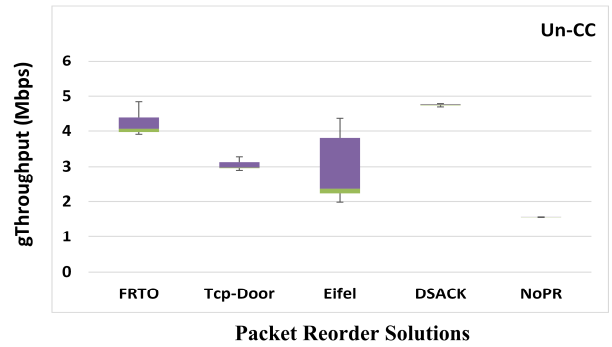(a) The gThroughput of MPTCP and Un-CC against PR solutions

(b) The gThroughput of MPTCP and Co-CC against PR solutions

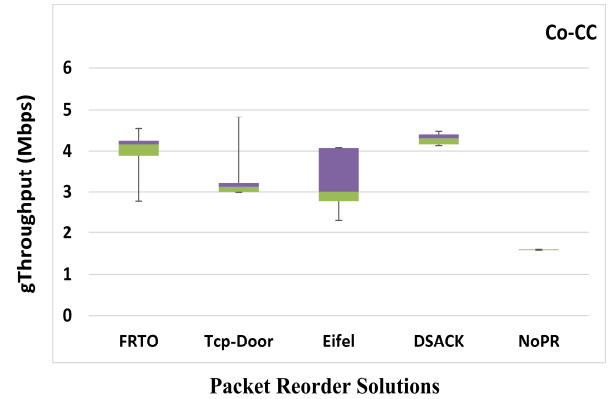Fig. 5. Scenario 3 gThroughput against PR solutions

By analysing the OOO-R and the buffer size results, D-SACK performance in terms of link utilization is similar to NoPR and Eifel because the transmission was through only one SF. However, the majority of D-SACK transmission is due to the use of the higher data rate link, whereas NoPR and Eifel use the slower link only. The OOO-R of both TCP-DOOR and F-RTO are larger than the others because their transmissions are through both links. TCP-DOOR requires 200 KB buffer in average for distances less than 50 m whilst the other comparable methods in this scenario require less. Refer to

Table III and Table IV as a typical set of results at distance 10 m.

Scenario 4 presents the case of a smaller difference in PHY data rates between different paths and uses a PHY data rate of 6 Mbps for 11g. In this scenario, both D-SACK and F-RTO provide gThroughput between 4 and 5 Mbps, whilst TCP-DOOR and Eifel provide less. NoPR depends only on the 11b SF which limits the gThroughput to be less than 2 Mbps. This behaviour is almost the same with all simulated distances and all CCs. However, the gThroughput of Eifel is 4 Mbps until a 40 m distance, and then gThroughput decreases dramatically by 1.5 Mbps. This justifies the variation in gThroughput of Eifel boxplot in Fig. 6 which illustrates the gThroughputs of this scenario.

(a) The gThroughput of MPTCP and Un-CC against PR solutions

(b) The gThroughput of MPTCP and Co-CC against PR solutions
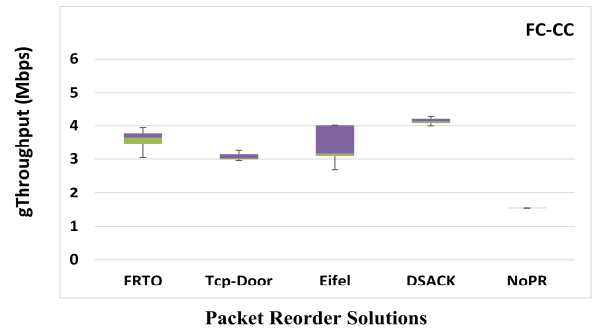
(c) The gThroughput of MPTCP and FC-CC against PR solutions

Fig. 6. Scenario 4 gThroughput against PR solutions

TABLE III.   OUT-OF-ORDER-RATIO FOR 10 M DISTANCE BETWEEN THE NODE AND THE WI-FI ACCESS POINT

Reordering Solution	Scenario-1			Scenario 2			Scenario 3			Scenario 4		
	OOO-R (%)			OOO-R (%)			OOO-R (%)			OOO-R (%)		
	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC
NoPR	1.3	1.8	1.3	4.5	5.7	4.3	3.9	5.4	3.8	4.0	5.5	3.9
DSACK	12.2	21.9	10.0	27.4	37.5	17.8	3.0	9.1	2.6	8.9	22.0	3.9
Eifel	2.1	10.1	1.7	5.4	11.3	5.3	8.8	13.2	6.3	2.5	17.2	2.4
TCP-DOOR	42.1	42.1	27.1	42.1	46.4	46.4	57.9	56.6	57.4	54.3	54.1	54.5
F-RTO	1.9	7.9	1.3	22.6	36.0	12.4	27.5	45.0	22.4	17.2	31.8	30.3

TABLE IV.   OUT-OF-ORDER BUFFER SIZE FOR 10 M DISTANCE BETWEEN THE NODE AND THE WI-FI ACCESS POINT

Reordering Solution	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	OOO Buffer Size (KB)			OOO Buffer Size (KB)			OOO Buffer Size (KB)			OOO Buffer Size (KB)		
	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC	Co-CC	Un-CC	FC-CC
NoPR	5.5	6.8	5.5	9.6	10.9	9.6	10.9	10.9	10.9	8.2	8.2	8.2
DSACK	30.1	30.1	76.6	21.9	21.9	32.8	23.2	140.8	21.9	15.0	27.3	9.6
Eifel	345.9	386.9	351.4	50.6	50.6	50.6	65.6	67.0	65.6	20.5	21.9	20.5
TCP-DOOR	299.4	438.9	336.3	157.2	131.3	101.2	113.5	166.8	131.3	102.5	160.0	218.8
F-RTO	57.4	56.1	23.2	53.3	49.2	16.4	56.1	49.2	42.4	41.0	67.0	71.1

Although both TCP-DOOR and F-RTO are not the best in their gThroughput, their OOO-R is larger than the others, due to their use of both wireless paths. D-SACK and Eifel in this scenario are the best in terms of gThroughput whilst also requiring less memory space.

V. DISCUSSION

The previous analysis shows that the Eifel solution does not provide any improvement in terms of gThroughput for the studied network scenarios, as concluded in [14], although for networks with heterogeneous paths, Eifel can still perform better than MPTCP-NoPR. This is because Eifel is a solution for situations where the wireless connectivity is intermittent, i.e., situations where spurious timeouts are likely to occur [18]. Since these situations do not exist in our scenarios frequently, Eifel does not provide any improvement for the studied scenarios. Adding to that, Eifel algorithm needs an appropriate link layer design to work well, which is out of the scope of this study.

MPTCP-DSACK and MPTCP-TCPDOOR perform well, particularly in networks with paths with similar characteristics (Scenario 1 & 2). This is because of how the sender of each technique responds to the received ACK signal. D-SACK uses SACK rather than the accumulative ACK, which is used by NoPR, to acknowledge discontinuous block of data. D-SACK increases CWND twice when an OOO packet is received. The receiver sends ACK with a DSACK option back to the sender when an OOO packet is received and stored in OOO buffer. It then sends another ACK later when the OOO packet stored in the OOO buffer becomes in order. Both cases allow the sender to increase CWND. The double increase in the CWND makes the related SF increase its transmission rate faster than the other SF, which makes the utilization of one SF better than the other. In TCP-DOOR, once the OOO is detected, the sender disables the congestion control and continue sending more data packets. Thus, no reduction to the CWND as NoPR dose by receiving the DUPACK.

In networks with dissimilar paths MPTCP-TCPDOOR is not in the top two best performance protocols. Both F-RTO and Eifel provide higher throughput than TCP-DOOR because the timeout events caused by OOO packets happen more frequently through these scenarios and F-RTO is able to detect the spurious timeouts.

The results also show that when the access network is wireless the coupling strategies have a small or no impact on the behaviour of the conventional MPTCP. This is different from studies with wired networks [14], and the main reason is that the difference between the end-to-end data paths is not in latency but only in the maximum supported data rate for each path in the wireless access network. However, this does not eliminate the importance of the coupling in terms of the fairness towards other flows sharing the path, which is out of the scope of this study. We believe that when a proper path selection strategy is applied to MPTCP then the effectiveness of the coupling methods will be more distinct even when the access network is wireless.

The memory requirements for OOO event processing are directly linked to the ability of the CC/PR method combination to use all SFs, as the incidence of OOO events increases clearly when both paths are used and more buffer space is needed to store OOO packets. It is worth mentioning that, from Scenario 1 and 3, we observed that gThroughput at distances more than 80m are less than 1 Mbps, while the other scenarios are not, which can be clearly observed from Fig. 3, Fig. 4, and Fig. 5. This is certainly related to the behaviour of the wireless physical link. When the 11g standard is operated at 64 QAM 3/4 MCS (modulation and coding scheme) to achieve a PHY data rate of 54 Mbps, the modulated signal becomes more affected by the low SINR at larger distances compared to the BPSK 1/2 MCS mode (6 Mbps). At low SINR levels (large distances), the receiver will not be able to decode the modulated signal at high modulation schemes (64 QAM 3/4), which results in high bit error rate and very low or no throughput.

VI. Conclusions

In this paper, different packet reorder recovery methods have been compared when run in conjunction with different congestion controllers for MPTCP over wireless networks where the link reliability and stability are the main differences between lossy and lossless networks. The study shows that the packet reordering solutions bring a substantial performance improvement for MPTCP by increasing the aggregate throughput as well as the path utilization. The MPTCP-DSACK provides the greatest number of instances with best performance and smallest OOO buffer footprint for different scenarios, and it is recommended for asymmetrical path networks whilst the MPTCP-TCPDOOR is the best for symmetrical path scenarios. Eifel packet reordering solution does not provide noticeable improvement in the aggregate throughput of MPTCP, which means it is not the best choice for throughput maximization, particularly for stabled connections, although it still provides connectivity redundancy. Furthermore, the congestion control strategies for MPTCP do not have a significant impact in cases when the latency is dominated by the core network rather than the access (wireless) network.

References

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

[2] H. Hsieh and R. Sivakumar, "pTCP: An End-to-End Transport Layer Protocol for Striped Connections," presented at the ICNP '02 Proceedings of the 10th IEEE International Conference on Network Protocols, 2002.

[3] A. Abd, T. Saadawi, and M. Lee, "LS-SCTP: A Bandwidth Aggregation Technique for Stream Control Ttransmission Protocol," *Computer Communications*, vol. 27, pp. 1012-1024, 2004.

[4] D. Sarkar, "A Concurrent Multipath TCP and its Markov Model," in *Communications, 2006. ICC'06. IEEE International Conference on*, 2006, pp. 615-620.

[5] 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*, 2007, pp. 135-142.

[6] J. Klei. (2014). *Continental to Showcase the Future of Connected Vehicles at the Connected Car Expo in L.A.* Available: http://www.theautochannel.com/news/2014/11/13/118993-continental-to-showcase-future-connected-vehicles-connected-car-expo-l.html

[7] Samsung. (2014). *Galaxy S5 Download Booster*. Available: http://galaxys5guide.com/samsung-galaxy-s5-features-explained/galaxy-s5-download-booster/

[8] D. Kaspar, K. Evensen, A. F. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and IP packet reordering over multiple wireless networks," in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, 2009, pp. 637-642.

[9] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A Measurement-based Study of Multipath TCP Performance over Wireless Networks," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 455-468.

[10] L. Ka-Cheong, V. O. K. Li, and Y. Daiqin, "An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, pp. 522-535, 2007.

[11] R. Costin, P. Christoph, B. Sebastien, F. Alan, H. Michio, D. Fabien, *et al.*, "How hard can it be? designing and implementing a deployable multipath TCP," in *NSDI'12 Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation.*, 2012, pp. 29-29.

[12] R. Costin, W. Damon, and H. Mark, "Practical Congestion Control for Multipath Transport Protocols," ed, 2009.

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

[14] A. Alheid, D. Kaleshi, and A. Doufexi, "An Analysis of the Impact of Out-of-Order Recovery Algorithms on MPTCP Throughput," in *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, 2014, pp. 156-163.

[15] A. Alheid, D. Kaleshi, and A. Doufexi, "Performance Evaluation of MPTCP in Indoor Heterogeneous Networks," presented at the Proceedings of the 2014 First International Conference on Systems Informatics, Modelling and Simulation, 2014.

[16] S. Barré, "Implementation and assessment of Modern Host-based Multipath Solutions," PhD Thesis, Université catholique de Louvain, 2011.

[17] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," *RFC2883*, 2000.

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

[19] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response," in *Proceedings of the 3rd ACM Interational Symposium on Mobile Ad Hoc Networking and Computing*, ed. Switzerland: ACM, 2002, pp. 217-225.

[20] P. Sarolahti, M. Kojo, and K. Yamamoto, "Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP," ed: RFC 5682, 2009.

[21] B. Chihani and D. Collange, "A Multipath TCP model for NS-3 simulator," in *Workshop on NS-3 held in conjunction with SIMUTools 2011*, 2011.

[22] *(2012) NS-3 website. [Online]. Available: http://www.nsnam.org/.*

[23] G. K. Karagiannidis, N. C. Sagias, and P. T. Mathiopoulos, "The N* Nakagami fading channel model," in *Wireless Communication Systems, 2005. 2nd International Symposium on*, 2005, pp. 185-189.