Experimental Analysis of Concurrent Multi-path Transmission Schemes

Varun Kumar Sharma[1], Lal Pratap Verma[2], Mahesh Kumar[3]
[1]Department of Computer Science and Engineering,The LNM Institute of Information Technology, Jaipur, Rajasthan, India
[2]Department of Computer Science and Engineering,Moradabad Institute of Technology, Moradabad, Uttar Pradesh, India
[3]Department of Computer Science and Engineering,Jaypee University of Engineering and Technology, Guna, Madhya Pradesh, India

Keywords: Concurrent Multi-Path Transmission, MPTCP, SCTP, Wired Networks.

Abstract: The mobile hosts having multi-interfacing capability can increase their performance (i.e., throughput) by effectively making use of concurrent transmissions over multiple available network paths. Nevertheless, this policy severely faces degradation in application-level throughput performance because of highly dissimilar characteristics (i.e., different delay and available bandwidth) of multiple network paths. In particular, the dissimilar paths' characteristics issue will persistently cause data to be received disordered (i.e., Out-Of-Order (OOO)) and due to negligibility of this issue, serious degradation in application-level throughput performance will occur. Consequently, these mentioned issues will further create the buffer blocking problem in the system and hence degrades the performance to a greater extent. In this paper, we evaluate and present an analysis of the concurrent transmission policies' performance over varying path characteristics. And we will understand what is the behaviour of the suggested concurrent transmission policies when the paths' characteristics are highly varied.

1 INTRODUCTION

The traditional Layer-4 (Transport Layer) protocols, TCP (Postel, 1981) and UDP (Postel, 1980), are fundamentally ignore the use of multi-homing feature. Specifically, TCP allows dynamic binding to only single network address at each end of the connection. At that time when TCP's functionality was suggested, the multi-interfacing capability were known to be inefficient because of expensive hardware requirements, hence, multi-homing feature was beyond the scope of interest for the researchers. Then, as the time passes, the desire for communication (data exchange) to be highly fault tolerant between end-to-end hosts, have conveyed multi-homing feature within the scope of interest for the researchers (Iyengar et al. 2006) (Sharma et al. 2019) (Verma and Kumar, 2017) (Verma et al., 2018) (Wallace and Shami, 2014)

The classic protocols engaged in heterogeneous network interface utilization are SCTP (Stewart et al., 2000) and MPTCP (Raiciu et al., 2011) (Ford et al., 2013) (Ford et al., 2011) (Paasch, and Bonaventure, 2014). In direction to transmission association establishment inclusive of numerous paths amid two end hosts, SCTP and its Concurrent Multi-path Transfer extension (CMT-SCTP) (Verma et al., 2018)exploit multi-homing. SCTP principally supports multi-homing competence and offers functionalities such as congestion and flow control, reliability and ordered data delivery (Natarajanet al., 2013). MPTCP offers the capability of concurrent utilization of numerous available network paths amid two end hosts, which has gained immense industrial attention and absorption from societies of research and standardized bodies (i.e., IEEE and IETF). In particular, the researchers have suggested the idea of combining the advantages of TCP and CMT in MPTCP. In this perspective, CMT dynamically exploits numerous available interfaces to effectively schedule data in concurrent fashion. Hence, CMT has an exceptional capability of providing significant fault tolerance, bandwidth aggregation and proper load balancing demands to multiple resource constraint applications. Nonetheless, the classical data transmission policies

Sharma, V., Verma, L. and Kumar, M.
Experimental Analysis of Concurrent Multi-path Transmission Schemes.
DOI: 10.5220/0010561600003161
In *Proceedings of the 3rd International Conference on Advanced Computing and Software Engineering (ICACSE 2021)*, pages 5-11
ISBN: 978-989-758-544-9

(e.g., CMT, CMT during Path-Failure (CMT-PF) (Natarajan et al., 2008)and CMT-SCTP) schedule the data over network paths in round-robin fashion. However, these approaches have not includes the path variable properties (e.g., available BW, path quality, and delay) of multiple network paths and transmit the data packet blindly. Consequently, these scheduling policies undeniably cause severe OOO data delivery at receiver which affects the performance significantly. Subsequently, OOO delivery causes unnecessary fast retransmissions and redundant congestion window (*cwnd*) reductions as well. Additionally, it leads to the issue of buffer-blocking problem in the network. The buffer-blocking problem severely restricts the possibility of data communication (exchange) and ultimately makes the connection idle, which subsequently degrades the performance in terms of average throughput and goodput respectively. Also, it leads to higher transmission delays and increases spurious retransmissions in the network.

The problem of redundant fast retransmissions can be dodged by designing and applying an efficient scheduling strategy. However, the most standard and common scheduling policy (i.e. round-robin scheduling policy) have not included the path dissimilar characteristics andschedules the transmission abruptly. Hence, the researchers have suggested numerous data scheduling policy (Sharma et al. 2019) (Verma and Kumar, 2017) (Verma et al., 2018)(Paasch, 2014) which effectively consider varying path characteristics, and hence reduces the re-ordering (packet) at the receiver's end. In this paper, we evaluate and present an analysis of the CMT, CMT-PF and MPTCP's performance over varying path characteristics.

The structure of this work as follows. Section 2 discusses the related works in the field of CMT. Section 3 discusses the simulation setup and environment used to test different CMT schemes. Section 4 discusses the experimental results. Section 5 finally concludes this work.

2 RELATED WORK

In order to describe about the foremost issues affecting several SCTP and MPTCP based proposals is the main objective of this section. Furthermore, we will look at what the researchers have suggested to reduce all the issues related with SCTP and MPTCP based proposals in this section.

2.1 SCTP based Proposals

SCTP does not support CMT; hence, Iyengar*et al.* (Iyengar et al. 2006) suggested an optimize version SCTP also known as CMT-SCTP. CMT-SCTP minimizes the reordering problem in CMT such as: superfluous fast retransmissions; SCTP sender wrongly interprets the reason for a packet loss, that is, network congestion. Consequently, SCTP sender needlessly reduce its *cwnd* size which majorly influences the network's average throughput performance. However, there could be another reason for the unordered packet, i.e., a packet could be delayed somewhere at the longer path. Moreover, excessive acknowledgment (ACK) traffic and receiver buffer blocking are other serious issues which are effectively handles by CMT-SCTP. Natarajan et al. (Natarajan et al., 2008) (Natarajan et al., 2013) have presented buffer-blocking problem associated with CMT, and authors show "how this problem severely hampers the performance of CMT during permanent (long) and short-term packet losses". Then, Verma and Kumar (Verma et al., 2018) investigated that the load distribution policy of CMT and CMT-PF blindly schedules the load over network paths without considering the path's bandwidth and delay variations. Hence, the authors have given an adaptive data chunk distribution policy for CMT (A-CMT) which effectively schedules the transmission load over paths considering both delay variations and available bandwidth factors. Still, an effective and deep evaluation of these above-mentioned approaches is indeed needed in highly lossy environment (e.g., Mobile Ad-hoc Networks (MANETs)). Hence, Xu *et al.* (Xu et al., 2013) have given Quality-Aware CMT (CMT-QA) for heterogeneous environment and extensively evaluates their approach compared to CMT and CMT-PF. Nevertheless, CMT-QA solely focuses on throughput performance enhancement and lacks concentration over fairness towards other competing TCP traffic flows. In addition, CMT-QA determines the packet losses only at Layer-4 level, which makes this approach highly inefficient in wireless environment, where appropriate buffer-overflow induced and channel characteristics induced loss classification (see details in (Sharma and Kumar, 2017) and references therein) is highly required. Also, many of the other proposals (Wallace and Shami, 2014) (Perotto et al., 2007), likewise CMT-QA, depends exclusively on Layer-4 Quality of Service (QoS) based parameters and also these policies do not consider other reasons for a packet drop apart from buffer overflow, hence, these policies suffer lower performance. Recently, Network-Coding (NC) based Layer-4 policies (Xu et al., 2015) (Xu et al. 2016) (Xu et al. 2017) have been proposed and has by now been confirmed a proficient method to solve buffer-blocking issue. With this we come to finish our short argument on suggested CMT solutions. Interested scholars and researchers can refer to Habib *et al*. (Habib et al. 2016), Xu *et al*. (Xu et al., 2016), Wallace and Shami(Wallace and Shami, 2012), Becke*et al*. (Becke et al., 2013) and Li *et al*. (Li et al., 2016)for further profound study on different CMT related solutions. Further interested researchers can refer to Sharma et al. (Sharma et al., 2019), Sharma and Kumar (Sharma and Kumar, 2017), Sharma et al. (Sharma et al, 2018)(Sharma and Kumar, 2017) (Sharma et al., 2018) (Sharma et al., 2012) (Sharma et al., 2020),Kanellopoulos and Sharma (Kanellopoulos and Sharma, 2020), and(Sharma, 2019)for congestion and energy aware solutions for single and multi-path Layer-4 protocols as well.

2.2 MPTCP based Proposals

The multi-pathing approach was inescapable and researchers were confident about that since architecturescontinuallysearching more feasible solutions. This, in return, resulted in MPTCP's policy development. Initially, numerous congestion control policies have been suggested which straightaway extended TCP NewReno for the purpose of designing MPTCP's policy (attributed as Linked Increase Algorithm (LIA) (Raiciu et al., 2011)), i.e., suggested policies directly triggers the TCP NewReno's functionality independently on each sub-path. This direct extended version can lead to severe un-fairness in the network for single-path TCP users when the obtainable network paths share bottleneck links with network paths used by MPTCP users. Hence, several researchers have suggested numerous mechanisms (Kelly and Voice, 2005) (Han et al., 2006) (Wang et al., 2003) with the intention of structuring an efficient multi-path Layer-4 protocol, specifically, compatible with the standard TCP (i.e., Coupled Congestion Control (CCC) algorithm). These suggested algorithms in (Kelly and Voice, 2005) (Han et al., 2006) (Wang et al., 2003) to utilize only the best available paths to the users and are best suited for the conditions where similar or little variations in Round Trip Time (RTTs) has been observed. Nevertheless, these algorithms suffer from the problem of flappiness (see details of flappiness and Opportunistic Linked-Increases Algorithm (OLIA) in (Khalili et al., 2013) and references therein) and lesser responsiveness. Firstly, these algorithms sometimes fail to adapt rapidly, in particular, they do not able to probe the paths with higher channel and congestion induced loss probabilities, hence, makes these algorithms much lesser responsive. Secondly, these algorithms show severe flappiness in the network. In particular, in order to resolve the lesser responsiveness issue associated with CCC algorithm Wischik*et al*. (Wischik et al., 2011) have suggested a novel Half-coupled congestion control mechanism which shows more responsiveness and it is more friendly towards single-path TCP users as well. Nevertheless, Peng *et al*. (Peng et al., 2013) (Peng et al., 2016) further claimed that Half-coupled congestion control mechanism un-friendliness towards single-path TCP users gets inflated during dissimilar RTT variations of each available sub-paths. Subsequently, the authors have identified design standard that give assurance of uniqueness, stability of system and existence. Their approach (attributed as Balanced Link Adaptation Algorithm (BALIA)) mainly focused on performance metrics such as receptiveness, TCP friendliness and window (congestion) variations. Singh *et al*. (Singh et al., 2013) have further claimed that there are still some performance issues have been associated with OLIA when all the available sub-paths are congested. They have suggested Adapted OLIA (AOLIA) policy which effectively controlled the aggressiveness of MPTCP in terms of *cwnd* growth scheme.

3 SIMULATION SETUP AND PARAMETERS

The simulation has been carried out on Network Simulator-2 (*ns*-2) with in-built MPTCP module originally modelled by Nishida (Nishida, 2013). The experiments considered the wired network environment shown in Fig. 1. As the figure demonstrates, end hosts 'S' and 'D' are attached with two interfaces 'S1–S2' and 'D1–D2' respectively whose configuration parameters are listed in Table 1. Whereas, end host 'S' is attached to single-homed routers 'R1' and 'R3', which further introduce heavy cross-traffic to simulate severe congested situation over network paths (PATH-1 and PATH-2). For this, the routers 'R1' and 'R3' are attached with UDP with Constant Bit Rate (CBR) traffic generating agent whose configuration parameters are listed in Table 1. In particular, there are two available network paths i.e., PATH-1 and PATH-2 with bottlenecks in the simulated environment. PATH–1's and PATH–2's bottleneck has 1 Mbps bandwidth and 45 ms propagation delay. Additionally, PATH–1 has 1% Packet Loss Rate (PLR) while PATH–2 has variable PLR which varies between 1%–10%. All the simulation results presented are estimated by normalizing the results over hundred runs, which makes the consequence of the loss rate and cross-traffic on simulated policies be more accurate and not effected by any other stochastic factors. Furthermore, all the necessary specific simulation parameters are listed in Table.1 below.

Table 1. Configuration Parameters.

Parameters	Values
MPTCP Maximum Segment Size (MSS)	1500 Bytes
MPTCP Sender Buffer Size	64 KB
MPTCP Receiver Buffer Size	64 KB
MPTCP Application	File Transfer Protocol (FTP)
SCTP MSS	1500 Bytes
SCTP Data Chunk Size	1468 Bytes
SCTP Sender Buffer Size	64 KB
SCTP Receiver Buffer Size	64 KB
SCTP RTX Policy	RTX-CWND
SCTP Application	FTP
Queuing Scheme	Drop Tail
Queue Size	50 Packets
Bottleneck Bandwidth	1 Mbps
Paths' Propagation delay	45 ms
Background Traffic	UDP
Simulation Period	200 sec.
UDP Application	PATH–1: 300 Kbps, PATH–2: 400 Kbps
PLR	PATH–1: 1%, PATH–2: 1%–10%

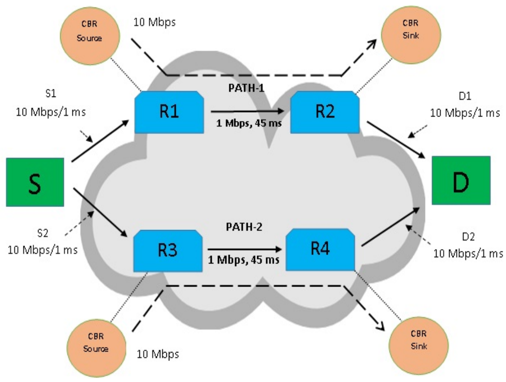

Figure 1: Topology

4 EXPERIMENTAL ANALYSIS

Fig. 2 exhibits the performance analysis in terms of the average throughput (Kbps) when the Packet Loss Rate (PLR) (%) increases. The objective of this experiment is to validate the capability of all the simulated CMT policy to deal with packet loss, which has considerable influence on average throughput performance. In fact, Fig. 2 shows the performance in terms of average throughput of all the simulated CMT. In particular, as the PLR increases, it radically enhances the chances of higher *cwnd* growth reductions. Moreover, it also increases the probability of higher transmission delay as well.

In particular, we can clearly observe that there is around 29.23%, 32.52% and 29.18% drop in throughput of MPTCP, CMT and CMT-PF respectively as the PLR varies between 1%–10%. This is due to the fact that all these approaches reduce their *cwnd*immediately as soon as they sense the packet loss. However, CMT-PF suggests improved performance because it can accurately recognize packet drop as a result of short-term route failures. Here, the results clearly signify that the overall throughput performance of MPTCP is pretty much lesser than that of other CMT approaches. Specifically, MPTCP's overall throughput performance is 19.50% less than that of CMT-PF and around 14.20% less than that of CMT. This is due to the fact that both CMT and CMT-PF use congestion control scheme independently for each available sub-paths, and hence it leads to high uncontrolled or aggressive *cwnd* growth behavior in the network. Hence, it certainly assists CMT and CMT-PF in terms of improved average throughput performance. While, MPTCP uses CCC algorithm which subsequently performs the congestion control considering the status of each available sub-paths, and hence it significantly controlled the aggressiveness of *cwnd* growth for each sub-path in the network. Hence, it leads to lesser average throughput performance in the system. Nevertheless, CMT and CMT-PF severely lacks of fairness against non-CMT users, while MPTCP is far more fair against single-path TCP users.

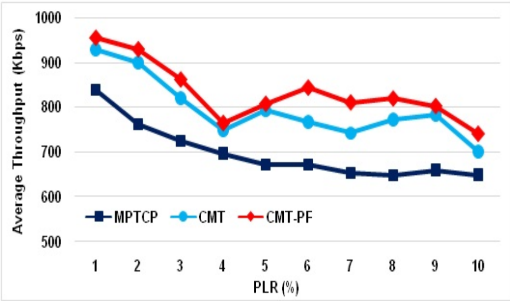

Figure 2: Average throughput (Kbps) performance of simulated CMT schemes for varied PLRs.

Fig. 3 illustrates the comparative analysis of the throughput (Kbps) when the bandwidth (Mbps) increases. The purpose of this experiment is to verify the competence of all the simulated CMT schemes on increasing bandwidth values. Specifically, we have simulated and analysed these results by keeping PATH–1 bandwidth value constant, while, we have varying the bandwidth values of PATH–2 in between 100 Kbps to 1.0 Mbps. Fig. 3 shows the average throughput performance of all the simulated CMT schemes continue to increase as bandwidth values continue to increase. In particular, at 100 Kbps bandwidth, the average throughput performance of MPTCP is around 22% more than that of CMT and approximately 4% more than that of CMT-PF. Also, MPTCP suggests comparable performance to CMT-PF at 200 Kbps bandwidth, while MPTCP offers 13% more average throughput performance as that of CMT. Meanwhile, similar average throughput performance has been observed for MPTCP than that of CMT and CMT-PF at 300 Kbps bandwidth. Here, when the available bandwidth of PATH–2 is limited (i.e., 100 Kbps to 300 Kbps); MPTCP's CCC algorithm effectively controls the *cwnd* growth aggressiveness on both available network sub-paths, and hence MPTCP's performance is better than that of CMT and comparable as that of CMT-PF. Since, CMT and CMT-PF's congestion control policy causes higher*cwnd* growth aggressiveness on both available network sub-paths, and hence their policy assist in sufficiently increasing the size of *cwnd*. Consequently, their average throughput performance is adequately better than that of MPTCP. Hence, on limited bandwidth values these policies are more likely to experience high packet losses and that subsequently reduces their average throughput performance. However, due to dependent congestion control policy (i.e., CCC Algorithm) limits MPTCP to aggressively utilize the available bandwidth, in particular, it fails in sufficiently increasing the size of *cwnd*, and hence MPTCP average throughput performance is less than that of other simulated policies. Specifically, MPTCP's average throughput performance is around 15% more than that of CMT and comparable performance has been observed with CMT-PF. While at higher bandwidth variations (i.e., 400 Kbps to 1 Mbps), MPTCP's average throughput performance is 23.22% and 19.78% less than that of CMT-PF and CMT respectively.

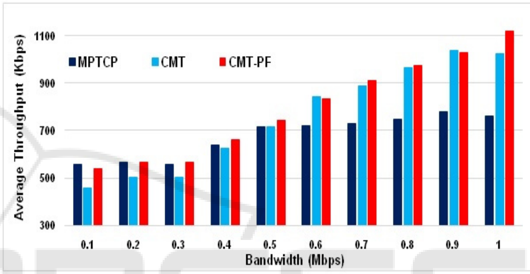

Figure 3: Average throughput (Kbps) performance of simulated CMT schemes for varied Bandwidth (Mbps).

Fig. 4 shows the comparative analysis of the throughput (Kbps) performance as the path delay (ms) increase. The purpose of this experiment is to verify the competence of all the simulated CMT schemes on increasing path delay values. Specifically, we have evaluated these results by keeping PATH–1 delay constant, while, varying the path delay of PATH–2 in between 10 ms to 100 ms. Here, it has been observed that around 11% and 12.37% drop in average throughput performance of CMT and CMT-PF respectively. Meanwhile, in case of MPTCP, serious drop in average throughput performance (i.e., around 54.45%) has been observed. This effect is any increase in path delay causes more OOO delivery at the end host (i.e., destination) which subsequently causes unnecessary fast retransmissions and redundant *cwnd* reductions. That ultimately leads to the issue of buffer-blocking problem and reduces the average throughput performance drastically. In particular, there is slight drop in average throughput performance has been observed for CMT and CMT-PF because their aggressive *cwnd*growth policy rapidly manages to utilize the available channel bandwidth well on time.

However, MPTCP lacks in effectively achieving the channel utilization due to its less aggressive *cwnd*growth policy, and hence its average throughput performance gets seriously affected.

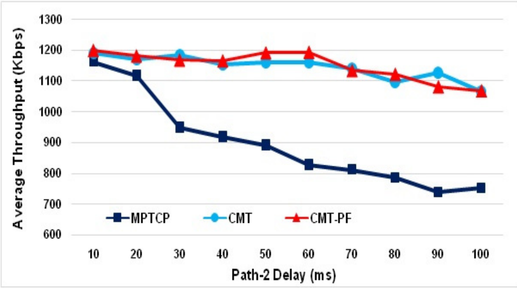

Figure 4. Average throughput (Kbps) performance of simulated CMT schemes for varied Path Delay (ms).

5 CONCLUSIONS AND FUTURE SCOPE

This paper evaluated and presented an analysis of the CMT, CMT-PF and MPTCP's performance over varying path characteristics. And along with that, we understood how dissimilar characteristics (i.e., PLR, bandwidth and path delay) of multiple available network paths made the difference to the old schemes given. We revealed that old data scheduling scheme (i.e., round-robin scheduling) and lesser aggressive *cwnd* growth behavior significantly affected MPTCP's average throughput performance. Since, both CMT and CMT-PF independently adapt their *cwnd*growth, hence, this aggressive *cwnd* growth behavior assists both policies in effectively utilizing channel bandwidth. Consequently, their average throughput performance is significantly more than that of MPTCP. Still, it is not rational to imply that we must go for only CMT and CMT-PF scheme but not for MPTCP. If we talk about fairness in particular, the CMT and CMT-PF policy significantly fails in achieving fairness to non-CMT users, while, MPTCP performs considerably well by achieving fairness to single-path TCP users.

The current suggested method makesutilization of all the accessible sub-flows. Still, there may be the possibility that better throughput performance may be achieved by dynamically eradicating based on their individual congestion. Furthermore, methodical and accurate assessment of the traffic scheduling and path management concern is unquestionably required prior to widespread deployment of these concurrent transmission polices in actual Internet environment.

REFERENCES

A. Singh, M. Xiang, A. Konsgen, C. Goerg, and Y. Zaki. (2013) "Enhancing fairness and congestion control in multipath TCP." In 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC), 1–8.

A.Ford, C.Raiciu, M.Handly, and O.Bonaventure. (2013) "TCP extensions for multipath operation with multiple addresses." IETF RFC 6824 (Technical Report), https://tools.ietf.org/html/rfc6824. Accessed on 12 October 2017.

A.Ford, C.Raiciu, M.Handly, S.Barre, and J.Iyengar. (2011) "Architectural guidelines for multipath TCP development." IETF RFC 6182 (Technical Report), https://tools.ietf.org/html/rfc6182. Accessed on 1 October 2016.

C.Paasch, and O.Bonaventure. (2014) "Multipath TCP." Communications of the ACM 57 (4): 51–57.

C.Paasch, S. Ferlin, O. Alay, and O.Bonaventure. (2014) "Experimental evaluation of multipath TCP schedulers." In: Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop, 27–32.

C.Raiciu, M.Handly, and D.Wischik. (2011) "Coupled congestion control for multipath transport protocols." IETF RFC 6356, https://tools.ietf.org/html/rfc6356. Accessed on 14 December 2018.

C.Xu, J.Zhao, and G-M.Muntean. (2016) "Congestion Control Design for Multipath Transport Protocols: A Survey." IEEE Communications Surveys & Tutorials 18 (4): 2948–2969.

C.Xu, P.Wang, X.Wei, and G-M.Muntean. (2017) "Pipeline network coding-based multipath data transfer in heterogeneous wireless networks." IEEE Transactions on Broadcasting 63 (2): 376–390.

C.Xu, T.Liu, J.Guan, H.Zhang, and G.Muntean. (2013) "CMT-QA: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks." IEEE Transactions on Mobile Computing 12 (11): 2193–2205.

C.Xu, Z.Li, J.Li, H.Zhang, and G-M.Muntean. (2015) "Cross-layer Fairness-driven Concurrent Multipath Video Delivery over Heterogeneous Wireless Networks." IEEE Transactions on Circuits and Systems for Video Technology 25 (7): 1175–1189.

C.Xu, Z.Li, L.Zhong, H.Zhang, and G-M.Muntean. (2016) "CMT-NC: Improving the Concurrent Multipath Transfer Performance using network coding in wireless networks." IEEE Transactions Vehicular Technology 65 (3): 1735–1751.

D. Kanellopoulos, V. K. Sharma. (2020) "Survey on power-aware optimization solutions for MANETs." Electronics 9: 1129.

D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. (2011) "Design, implementation and evaluation of congestion control for multipath TCP." In: Proceedings of the 8th USENIX conference on Networked systems design and implementation, 99–112.

F. Kelly, and T. Voice. (2005) "Stability of end-to-end algorithms for joint routing and rate control." ACM

SIGCOMM Computer Communication Review 35 (2): 5–12.

F.Perotto, C.Casetti, and G.Galante. (2007) "SCTP-based Transport Protocols for Concurrent Multipath Transfer." In IEEE Wireless Communications and Networking Conference, 2969–2974.

H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. (2006) "Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet." IEEE/ACM Transactions on Networking 14 (6): 1260–1271.

J. Postel (1980) "User datagram protocol." IETF RFC 768, https://tools.ietf.org/html/rfc768. Accessed on 15 May 2019.

J. Postel (1981) "Transmission control protocol." IETF RFC 793, https://tools.ietf.org/html/rfc793. Accessed on 15 May 2019.

J.R.Iyengar, P.D.Amer, and R.Stewart. (2006) "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths." IEEE/ACM Transactions on Networking 14 (5): 951–964.

L.P.Verma, and M.Kumar. (2017) "An adaptive data chunk scheduling for concurrent multipath transfer." Computer Standard and Interfaces 52: 97–104.

L.P.Verma, V.K.Sharma, and M.Kumar. (2018) "New delay-based fast retransmission policy for CMT-SCTP." International Journal of Intelligent Systems and Applications 10 (3): 59–66.

M. Becke, H. Adhari, E. P. Rathgeb, F. Fa, X. Yang, and X. Zhou. (2013) "Comparison of Multipath TCP and CMT-SCTP based on intercontinental measurements." In: IEEE Global Communications Conference (GLOBECOM), 1360–1366.

M. Li, A. Lukyanenko, Z. Ou, A. Yla-Jaaski, S. Tarkoma, M. Coudron, and S. Secci. (2016) "Multipath Transmission for the Internet: A Survey," IEEE Communications Surveys & Tutorials 18 (4): 2887–2925.

P.Natarajan, N.Ekiz, P.D.Amer, and R.Stewart. (2009) "Concurrent multipath transfer during path failure." Computer Network 32 (15): 1577–1587.

P.Natarajan, N.Ekiz, P.D.Amer, J.R.Iyengar, and R.Stewart. (2008) "Concurrent multipath transfer using SCTP multihoming: Introducing the potentially-failed destination state." in Das A., Pung H.K., Lee F.B.S., Wong L.W.C. (eds) NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, 727–734.

Q. Peng, A. Walid, and S. H. Low. (2013) "Multipath TCP algorithms: theory and design." In: Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems, 305–316.

Q. Peng, A. Walid, J. Hwang, and S. H. Low. (2016) "Multipath TCP: Analysis, Design, and Implementation." IEEE/ACM Transactions on Networking 24 (1): 596–609.

R. Khalili, N. Gast, M. Popovic, and J-Y. Le Boudec. (2013) "MPTCP is not Pareto-optimal: performance issues and a possible solution." IEEE/ACM Transactions on Networking 21 (5): 1651–1665.

R.Stewart, Q.Xie, K.Morneault, C.Sharp, H.Schwarzbauer, T.Taylor, I.Rytina, M.Kalla, L.Zhang, and V.Paxson. (2000) "Stream control transmission protocol." IETF RFC 2960, https://tools.ietf.org/html/rfc2960. Accessed on 26 January 2019.

S.Habib, J.Qadir, A.Ali, D.Habib, M.Li, and A.Sathiaseelan. (2016) "The past, present, and future of transport-layer multipath." Journal of Network and Computer Applications 75: 236–258.

T.Wallace, and A.Shami. (2012) "A Review of Multihoming Issues Using the Stream Control Transmission Protocol." IEEE Communications Surveys & Tutorials 14 (2): 565–578.

T.Wallace, and A.Shami. (2014) "Concurrent multipath transfer using SCTP: Modelling and congestion window management." IEEE Transactions on Mobile Computing 13 (11): 2510–2523.

V. K. Sharma, and M. Kumar. (2017) "Adaptive Energy Efficient Load Distribution Using Fuzzy Approach." Ad Hoc & Sensor Wireless Networks 39 (1-4): 123–166.

V. K. Sharma, and M. Kumar. (2018) "Adaptive load distribution approach based on congestion control scheme in ad-hoc networks." International Journal of Electronics 106 (1): 48–68.

V. K. Sharma, S. S. P. Shukla, and V. Singh. (2012) "A Tailored Q-Learning for Routing in Wireless Sensor Networks." In: IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), 663–668.

V. K. Sharma. (2019) "Energy and congestion conscious transmissions and routing in SANETs and MANETs: A Survey." Engineering and Technology Journal for Research and Innovation 1(2): 38-42.

V.K.Sharma, and M.Kumar. (2017) "Adaptive congestion control scheme in mobile ad-hoc networks." Peer-to-Peer Networking and Applications 10 (3): 633–657.

V.K.Sharma, L.P.Verma, and M.Kumar. (2018) "A Fuzzy-based Adaptive Energy Efficient Load Distribution Scheme in Ad-hoc Networks." International Journal of Intelligent Systems and Applications 10 (2): 72–84.

V.K.Sharma, L.P.Verma, and M.Kumar. (2019) "CL-ADSP: Cross-layer adaptive data scheduling policy in mobile ad-hoc networks." Future Generation Computer Systems 97: 530–563.

V.K.Sharma, L.P.Verma, M.Kumar, R. K. Naha, A. Mahanti. (2020) "A-CAFDSP: An Adaptive-congestion aware Fibonacci sequence based data scheduling policy." Computer Communications 158: 141–165.

W-H Wang, M. Palaniswami, and S. H. Low. (2003) "Optimal flow control and routing in multi-path networks." Performance Evaluation 52 (2-3): 119–132.

Y. Nishida. (2013) "MPTCP implementation on NS-2." with IETF and WIDE Project, http://code.google.com/p/multipath-tcp. Accessed on 11 May 2017.